

---

# CoCon API Description

---

## Copyright

All information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Televic.

© 2018 Televic NV. All rights reserved.

## Document history

Version	Author	Date	Description
0.01	JV	23/01/2014	Initial version
0.02	DL	12/02/2014	Add commands description
0.03	DL	13/02/2014	Add badge event and delegate event
0.04	DL	07/03/2014	Add user information during microphone state change
1.0	JV	22/05/2014	Finalized the API for CoCon v3.1
1.1	JV	16/06/2014	Added commands for individual voting results
1.2	JV	24/06/2014	Defined the event model and commands
1.3	JV	11/07/2014	Added the "ButtonLED_Event" model
1.4	JV	18/08/2014	Added commands for <ul style="list-style-type: none"><li>- Interactive model: Delegate screen control</li><li>- Start meeting from XML file</li><li>- Agenda and voting control</li></ul>
1.5	MP	31/10/2014	Added and improved commands in the Meeting_Agenda model
1.6	GL	25/11/2014	Modify SeatTimerPauseResumed name and parameters
1.7	GL	26/12/2014	Added commands for

			<ul style="list-style-type: none"> <li>- ClearAllDelegatesAndGroups</li> <li>- StartEmptyMeeting</li> <li>- About Audio application commands</li> </ul>
1.8	GL	12/01/2015	<p>Added commands for</p> <ul style="list-style-type: none"> <li>- Initialization</li> </ul>
1.9	GL/MP	21/04/2015	<p>Added command for</p> <ul style="list-style-type: none"> <li>- SetNextButtonLedStatus</li> <li>- SetPriorButtonLedStatus</li> <li>- NextButtonPressedEvent</li> <li>- PriorButtonPressedEvent</li> <li>- Send message to Operator</li> </ul>
2.0	GL/MP	04/05/2015	<p>Agenda item ID replaced with sequence number ID is not well readable for user. But Sequence number is string ,It might be “1”, “1.1”, “1.1.2”</p> <p>Added commands for</p> <ul style="list-style-type: none"> <li>- RemoveAgendaItemById</li> <li>- DeleteAllAgendaItems</li> <li>- MoveAgendaItem</li> </ul>
2.1	GL/MP	21/05/2015	Added commands for Interpretation
4.1	KMA/MP	25/01/2016	<p>Updated with commands for</p> <ul style="list-style-type: none"> <li>- Interpreter</li> <li>- Intercom</li> <li>- Flexible delegate</li> <li>- Timer</li> </ul>
4.5	KMA	28/08/2016	<p>Updated commands for</p> <ul style="list-style-type: none"> <li>- Room (initialization)</li> </ul>
5.0	MP/KMA	03/04/2017	<p>Update API for</p> <ul style="list-style-type: none"> <li>- Timer setting enabled / disabled</li> <li>- Video application</li> <li>- API for microphone activation type</li> <li>- GetUnitErrorState</li> <li>- GetMatrixNodeStatus</li> <li>- GetCUState</li> <li>- GetNetworkDeviceList</li> </ul>
5.1	MP/KMA	12/07/2017	<p>Update API for:</p> <ul style="list-style-type: none"> <li>- StartEmptyMeeting (With login method as optional)</li> <li>- GetCulpAddress</li> <li>- SetCulpAddress</li> <li>- RestartRoomServer</li> <li>- GetAmountOfVotes</li> <li>- NotifyExternalCall</li> <li>- Intercom Parameter name improved. (Request/ Accept /End call )</li> <li>- AddSeat (New API)</li> <li>- RemoveSeat (New API)</li> <li>- AddChannel / EditChannel (add new para : IsRecording)</li> </ul>

5.1.1	MP	11/07/2017	<ul style="list-style-type: none"> <li>- Audio preset commands. (Newly added)</li> <li>- Interpreter Preset API improved with new optional parameter.</li> <li>- Remove unit from room.</li> <li>- SaveAndSendInterpreterPreset</li> <li>- AddDelegatesToMeeting</li> <li>- AddDelegatesToSeats</li> <li>- EditSeat &amp; EditBooth (intercom new field)</li> <li>- Audio API update events (MatrixNodeUpdated; ActiveAudioPresetUpdated; MatrixNodeVolumeUpdated)</li> <li>- API name correction and keep old API for exiting customer. (ActivateNextMicrophone; GetAudioConfigurationDetailById; GetAudioConfigurationDetailByName)</li> <li>- SetMicrophoneSensitivity / MicrophoneSensitivityUpdated</li> <li>- SetDelegateTimer &amp; SetGroupTimer update event added with the correct name.</li> </ul>
5.2	MP	12/20/2017	<ul style="list-style-type: none"> <li>- AddSynoptic</li> <li>- EditSeat (IsChairman)</li> <li>- GetDefaultMicrophoneModeForRoom</li> <li>- SetDefaultMicrophoneModeForRoom</li> <li>- DefaultMicrophoneModeForRoomUpdated</li> <li>- ClearAllMeetingAndDelegateData</li> <li>- AddVotingTemplate</li> </ul>
5.3.2	MP	09/10/2018	<ul style="list-style-type: none"> <li>- ResetDelegateTimer</li> <li>- ResetGlobalDelegateTimer</li> <li>- ResetGroupTimer</li> <li>- ResetGlobalGroupTimer</li> <li>- ChangeDelegateTimer</li> <li>- ChangeGroupTimer</li> <li>- ResendMeetingData</li> <li>- keypad_login_code (notification api)</li> </ul>
5.4	MP	05/12/2018	<ul style="list-style-type: none"> <li>- Updated api <ol style="list-style-type: none"> <li>1. Voting template with new para "IsUnanimous"</li> <li>2. AddDelegate, EditDelegate, DelegateAdded &amp; Delegateupdated api calls with new Para"MiddleName", "FingerprinterData", "KeypadLoginCode"</li> </ol> </li> <li>- Newly added api <ol style="list-style-type: none"> <li>1. SetAudioSoftwareConfigurationById (update notification)</li> <li>2. CurrentAudioSoftwareConfigurationUpdated(uptate notification)</li> <li>3. GetGeneralVotingResults</li> <li>4. Recording Module <ol style="list-style-type: none"> <li>i. GetRecordingFilesInfo</li> <li>ii. Startrecording</li> </ol> </li> </ol> </li> </ul>

			<ul style="list-style-type: none"> <li>iii. Stoprecording</li> <li>iv. Getrecordingstate</li> </ul> <p>5. Wireless Coupling Module</p> <ul style="list-style-type: none"> <li>i. Getwcapinfo</li> <li>ii. Getallwcapinfo</li> <li>iii. SetWcapCoupledMode</li> </ul>
5.4.1	MP	04/17/2019	<ul style="list-style-type: none"> <li>- API updated : <ul style="list-style-type: none"> <li>1. ClearAllMeetingAndDelegateData with optional parameter "IncludeService=True/False"</li> </ul> </li> </ul>
6.0	JHO	05/20/2019	<ul style="list-style-type: none"> <li>- API updated : CoCon for Plixus no longer supported</li> </ul>

## Table of contents

### Table of Contents

1	Introduction & Scope .....	13
1.1	Introduction .....	13
1.2	Scope.....	13
1.2.1	In scope .....	13
1.2.2	Out of scope .....	13
1.3	API module compatibility .....	13
	Server to Client .....	13
	Client to server .....	16
2	General communication structure .....	22
2.1	Communication format .....	22
2.1.1	REST/JSON.....	22
2.1.2	Long polling .....	22
2.2	Communication methods.....	22
2.3	Publish/Subscribe .....	23
2.4	Setting up and operating the CoCon API connection.....	24
2.4.1	General .....	24
2.4.2	REST/JSON Example.....	25
2.4.3	Server return values .....	26
3	Operational commands .....	27
3.1	Data models.....	27
3.1.1.1	AuthorityAssigned.....	27
3.1.1.2	Meeting .....	27
3.1.1.3	VotingCountWeight.....	27
3.1.1.4	VotingOption .....	27
3.1.1.5	VotingOptionVoteDetail .....	28
3.1.1.6	VotingResultSummary .....	28
3.1.1.7	IndividualVote .....	28
3.1.1.8	Timer.....	28
3.1.1.9	BaseObjectTimer .....	28
3.1.1.10	DelegateTimer : BaseObjectTimer .....	29
3.1.1.11	GroupTimer : BaseObjectTimer.....	29
3.1.1.12	SeatTimer : BaseObjectTimer .....	29
3.1.1.13	Delegate .....	29
3.1.1.14	Group .....	30
3.1.1.15	AgendaItem .....	30
3.1.1.16	VotingAgendaItem:AgendaItem.....	30
3.1.1.17	LecturerAgendaItem:AgendaItem.....	30
3.1.1.18	AudioGroup.....	30
3.1.1.19	AudioMatrixNode .....	30
3.1.1.20	AudioSoftwareConfiguration .....	31
3.1.1.21	AudioSoftwareConfigurationDetailed.....	31
3.1.1.22	ScreenControlOptions .....	31
3.1.1.23	HomeControl : ScreenControlOptions .....	31
3.1.1.24	DiscussionControl : ScreenControlOptions .....	31

3.1.1.25	AgendaControl : ScreenControlOptions .....	31
3.1.1.26	DelegatesControl : ScreenControlOptions .....	32
3.1.1.27	VotingControl : ScreenControlOptions .....	32
3.1.1.28	DocumentsControl : ScreenControlOptions .....	32
3.1.1.29	MessagesControl : ScreenControlOptions .....	32
3.1.1.30	ServicesControl : ScreenControlOptions .....	32
3.1.1.31	ConfigurationControl : ScreenControlOptions .....	32
3.1.1.32	Unit.....	32
3.1.1.33	Channel.....	32
3.1.1.34	InterpreterPreset.....	33
3.1.1.35	DeskUnitInChannel.....	33
3.1.1.36	ChannelRealtimeData.....	33
3.2	Server to client.....	33
3.2.1	Room .....	34
3.2.1.1	InitializationState.....	34
3.2.1.2	UnitStateChanged .....	34
3.2.1.3	VolumeChangedForRoom.....	34
3.2.1.4	UnitError .....	35
3.2.2	Microphone .....	35
3.2.2.1	MicrophoneState.....	35
3.2.2.2	MicrophoneMode.....	36
3.2.2.3	MicButtonEvent.....	37
3.2.2.4	MicrophoneSensitivityUpdated .....	37
3.2.2.5	MicStateChangedInRunningMeeting.....	37
3.2.3	Meeting_Agenda.....	38
3.2.3.1	MeetingTitleChanged.....	38
3.2.3.2	MeetingStatus.....	38
3.2.3.3	MeetingAgendaChanged.....	39
3.2.3.4	Agenda_ItemChanged.....	39
3.2.3.5	Keypad_login_code .....	40
3.2.4	Voting.....	40
3.2.4.1	VotingState .....	40
3.2.4.2	VotingOutcome .....	40
3.2.4.3	GeneralVotingResults.....	41
3.2.4.4	IndividualVotingResults .....	41
3.2.4.5	VotingTemplateAdded .....	42
3.2.5	Timer.....	47
3.2.5.1	MeetingTimerEvent.....	47
3.2.5.2	MeetingTimerStateChange.....	47
3.2.5.3	DelegateTimerEvent .....	48
3.2.5.4	DelegateTimerStateChange .....	48
3.2.5.5	GroupTimerEvent .....	48
3.2.5.6	GroupTimerStateChange .....	49
3.2.5.7	SeatTimerEvent.....	49

3.2.5.8	SeatTimerPauseResumed .....	50
3.2.5.9	ActiveAgendaItemTimerEvent .....	50
3.2.5.10	ActiveAgendaItemTimerPauseResumed.....	50
3.2.5.11	VotingTimerEvent .....	51
3.2.5.12	VotingTimerPauseResumed.....	51
3.2.6	Delegate .....	52
3.2.6.1	BadgeEvent .....	52
3.2.6.2	DelegateOnSeat .....	52
3.2.6.3	MeetingStartsUserInformation .....	53
3.2.6.4	DelegateUpdated.....	53
3.2.6.5	DelegateAdded .....	54
3.2.6.6	DelegateRemoved .....	55
3.2.6.7	DelegateSeatingUpdated .....	55
3.2.6.8	DelegatesInMeetingUpdated .....	56
3.2.7	Interactive .....	56
3.2.7.1	ScreenLockRemoved .....	56
3.2.7.2	DelegateScreenSet.....	56
3.2.7.3	CustomButtonEvent.....	57
3.2.8	Audio.....	57
3.2.8.1	SetAudioSoftwareConfigurationByIid .....	57
3.2.8.2	CurrentAudioSoftwareConfigurationUpdated .....	58
3.2.9	Logging .....	58
3.2.9.1	ConnectionLost_CentralUnit.....	58
3.2.9.2	ConnectionLost_Database .....	58
3.2.10	ButtonLED_Event .....	59
3.2.10.1	ButtonPressed .....	59
3.2.10.2	NextButtonPressed .....	59
3.2.10.3	PriorButtonPressed.....	60
3.2.11	Interpretation.....	60
3.2.11.1	IMicrophoneStateUpdated .....	60
3.2.11.2	IActivatedChannelUpdated .....	60
3.2.11.3	SpeakSlowerRequest .....	61
3.2.11.4	SpeakSlowerEnded .....	61
3.2.11.5	ActivePresetStateChanged .....	61
3.2.11.6	.....	61
3.2.12	Recording ( AE-R & Plixus Core only).....	61
3.2.12.1	RecordingStatusUpdate.....	61
3.2.13	Wireless Coupling.....	62
3.2.13.1	WcapInfoUpdated .....	62
3.2.13.2	WcapCoupledUpdated.....	62
3.2.13.3	AllWcapInfoUpdated .....	62
3.3	Client to server.....	63
3.3.1	General .....	63
3.3.1.1	Connect .....	63
3.3.1.2	Subscribe.....	63

3.3.1.3	Unsubscribe .....	63
3.3.1.4	Notification .....	64
3.3.1.5	RebootSystem .....	64
3.3.2	Room .....	65
3.3.2.1	GetRoomServerName .....	65
3.3.2.2	GetActiveRoomInfo.....	65
3.3.2.3	GetStatusOfCuAndDbConnection .....	66
3.3.2.4	SetInitializationState .....	66
3.3.2.5	GetAllUnitsState .....	66
3.3.2.6	SetDefaultVolumeForRoom.....	67
3.3.2.7	GetDefaultMicrophoneModeForRoom.....	67
3.3.2.8	SetDefaultMicrophoneModeForRoom .....	68
3.3.2.9	SetVolumeForRoom .....	69
3.3.2.10	AddSynoptic.....	69
3.3.2.11	GetAllSeats .....	70
3.3.2.12	GetBooths .....	70
3.3.2.13	EditSeat .....	70
3.3.2.14	GetAllUnits .....	71
3.3.2.15	SetOperatingMode.....	72
3.3.2.16	GetOperatingMode .....	72
3.3.2.17	ClearAllMeetingAndDelegateData.....	73
3.3.3	Microphone .....	73
3.3.3.1	SetState .....	73
3.3.3.2	SetMicrophoneMode.....	74
3.3.3.3	SetActivationType.....	75
3.3.3.4	SetMicrophoneSensitivity .....	75
3.3.3.5	Get .....	76
3.3.3.6	GetState.....	76
3.3.3.7	ClearMicrophoneList.....	76
3.3.3.8	GetMicrophoneMode .....	77
3.3.3.9	ActivateNextMicrophone.....	77
3.3.4	Meeting_Agenda.....	77
3.3.4.1	SetMeetingState .....	77
3.3.4.2	StartEmptyMeeting .....	78
3.3.4.3	GetMeetingsForToday .....	78
3.3.4.4	GetAllMeetingsForDate .....	79
3.3.4.5	GetMeetingById .....	79
3.3.4.6	SetActiveAgendaItemByDirection.....	80
3.3.4.7	SetActiveAgendaItemById .....	80
3.3.4.8	GetAgendaItemInformationInRunningMeeting .....	80
3.3.4.9	GetAgendaItemInformationInMeetingById .....	81
3.3.4.10	GetAgendaItemById .....	82
3.3.4.11	StartMeetingViaMeetingTemplate .....	82



3.3.4.12	StartMeetingViaXMLimport.....	82
3.3.4.13	AddAgendaItem .....	83
3.3.4.14	GetLoginMethodForMeeting .....	84
3.3.4.15	RemoveAgendaItemById.....	84
3.3.4.16	DeleteAllAgendaItems .....	85
3.3.4.17	MoveAgendaItem.....	85
3.3.4.18	EditAgendaItem .....	86
3.3.4.19	AddDelegateToMeeting .....	86
3.3.4.20	AddDelegatesToMeeting .....	87
3.3.4.21	RemoveDelegateFromMeeting.....	87
3.3.4.22	GetDelegateSeating .....	88
3.3.4.23	AddDelegateToSeat .....	88
3.3.4.24	AddDelegatesToSeats.....	89
3.3.4.25	RemoveDelegateFromSeat .....	89
3.3.4.26	ApplyDelegatesToSystem .....	90
3.3.4.27	EndActiveMeeting.....	90
3.3.4.28	ResendMeetingData .....	90
3.3.5	Voting.....	91
3.3.5.1	SetVotingState .....	91
3.3.5.2	AddInstantVote .....	92
3.3.5.3	AddVotingTemplate .....	92
3.3.5.4	GetVotingState .....	101
3.3.5.5	GetAmountOfVotes.....	101
3.3.5.6	GetGeneralVotingResults.....	101
3.3.6	Timer.....	102
3.3.6.1	Get .....	102
3.3.6.2	GetDelegateTimers.....	102
3.3.6.3	GetSeatTimers.....	103
3.3.6.4	GetGroupTimers .....	103
3.3.6.5	GetActiveTimers .....	103
3.3.6.6	SetDelegateTimer.....	103
3.3.6.7	SetGroupTimer .....	104
3.3.6.8	SetGlobalDelegateTimer .....	105
3.3.6.9	SetGlobalSeatTimer .....	106
3.3.6.10	SetGlobalGroupTimer.....	106
3.3.6.11	ResetDelegateTimer.....	107
3.3.6.12	ResetGlobalDelegateTimer .....	108
3.3.6.13	ResetGroupTimer .....	108
3.3.6.14	ResetGlobalGroupTimer.....	108
3.3.6.15	ChangeDelegateTimer.....	109
3.3.6.16	ChangeGroupTimer .....	109
3.3.7	Delegate .....	110
3.3.7.1	GetDelegatesInMeeting.....	110

3.3.7.2	GetDelegatesInCurrentMeeting .....	111
3.3.7.3	ClearAllDelegatesAndGroups .....	111
3.3.7.4	GetAllDelegates .....	112
3.3.7.5	AddDelegate .....	112
3.3.7.6	RemoveDelegate .....	114
3.3.7.7	EditDelegate .....	114
3.3.7.8	ImportDelegatesFromFile .....	115
3.3.8	Interactive .....	115
3.3.8.1	SetDelegateScreen .....	116
3.3.8.2	RemoveScreenLock .....	119
3.3.8.3	SetCustomButtonOnSeat .....	120
3.3.8.4	SetCustomButtonOnRoom .....	120
3.3.8.5	SetCustomLabelOnSeat .....	120
3.3.8.6	SetCustomLabelOnRoom .....	121
3.3.8.7	SetCustomScreenOnSeat .....	121
3.3.8.8	SetCustomScreenOnRoom .....	122
3.3.9	Audio .....	122
3.3.9.1	GetCurrentAudioSoftwareConfiguration .....	122
3.3.9.2	GetCurrentAudioSoftwareConfigurationInDetails .....	122
3.3.9.3	GetAudioSoftwareConfigurations .....	128
3.3.9.4	GetAudioGroupsInCurrentConfiguration .....	129
3.3.9.5	SetAudioSoftwareConfigurationById .....	129
3.3.9.6	SetAudioSoftwareConfigurationByName .....	130
3.3.9.7	ChangeMatrixNodeStatus .....	130
3.3.9.8	GetMatrixNodeStatus .....	131
3.3.9.9	SaveAudioSoftwareConfiguration .....	131
3.3.10	Logging .....	132
3.3.10.1	GetCUState .....	132
3.3.10.2	GetUnitErrorState .....	132
3.3.11	ButtonLED_Event .....	132
3.3.11.1	SetButtonsEnabled .....	132
3.3.11.2	SetLED .....	133
3.3.11.3	SetNextButtonEnabled .....	134
3.3.11.4	SetPriorButtonEnabled .....	134
3.3.11.5	SetNextButtonLedStatus .....	135
3.3.11.6	SetPriorButtonLedStatus .....	135
3.3.12	External .....	136
3.3.12.1	Send message to Operator .....	136
3.3.12.2	NotifyExternalCall .....	136
3.3.13	Interpretation .....	137
3.3.13.1	GetAllBoothsInRoom .....	137
3.3.13.2	AddLanguage .....	137
3.3.13.3	RemoveLanguage .....	137
3.3.13.4	ClearLanguages .....	138

3.3.13.5	AddInterpreterPreset .....	138
3.3.13.6	DeleteInterpreterPreset .....	139
3.3.13.7	ApplyInterpreterPreset.....	140
3.3.13.8	GetActiveInterpreterPreset .....	140
3.3.13.9	GetAllInterpreterPresets .....	141
3.3.13.10	GetAllLanguages .....	141
3.3.13.11	GetAllChannels .....	141
3.3.13.12	AddChannel .....	142
3.3.13.13	EditChannel .....	143
3.3.13.14	DeleteChannel .....	144
3.3.13.15	RemoveAllChannels .....	144
3.3.13.16	SetDisplayText.....	145
3.3.13.17	SaveInterpreterPresetAs .....	145
3.3.13.18	GetAllChannelRealtimeData.....	147
3.3.13.19	GetSpeakSlowerRequestList.....	147
3.3.13.20	GetInterpreterPresetById.....	148
3.3.14	API calls for Configurations .....	148
3.3.14.1	GetCulpAddress .....	148
3.3.14.2	SetCulpAddress.....	148
3.3.14.3	RestartRoomServer .....	149
3.3.14.4	GetCoconServerVersion.....	149
3.3.15	Recording ( AE-R & Plixus Core only).....	149
3.3.15.1	Getrecordingfilesinfo.....	149
3.3.15.2	Startrecording .....	150
3.3.15.3	Stoprecording .....	151
3.3.15.4	Getrecordingstate .....	151
3.3.16	Wireless coupling.....	151
3.3.16.1	Getwcapinfo.....	151
3.3.16.2	Getallwcapinfo .....	152
3.3.16.3	SetWcapCoupledMode.....	152
4	Future commands .....	152
4.1	Server to client.....	152
4.2	Client to server.....	152

**List of figures**

No table of figures entries found.

**List of tables**

No table of figures entries found.

**Terminology**

Name	Meaning

**References**

ID	Reference	Version	Name and meaning

# 1 Introduction & Scope

## 1.1 Introduction

This document contains the description of the API (Application Programmers' Interface) of the CoCon Room Server. Through this API, CoCon offers two functionalities:

- Publication of events that are happening in the conference room and software;
- External control of the conference room and software.

## 1.2 Scope

### 1.2.1 In scope

The current version of the document targets one technology that is used for the API. This is REST (Representational State Transfer) with JSON format.

The general working of this system is described and the commands are defined and shown with examples.

### 1.2.2 Out of scope

- Other technologies than REST/JSON.
- Other external points of access to the CoCon software and Televic Conference products in general.

## 1.3 API module compatibility

This table shows which API modules are available in the different systems.

API Module	Cocon for Plixus Core	Plixus Core
<b><u>Server to Client</u></b>		
<b><u>3.2.1 Room</u></b>		
3.2.1.1 InitializationState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.1.2 UnitStateChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.1.3 VolumeChangedForRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.1.4 UnitError	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.2 Microphone</u></b>		
3.2.2.1 MicrophoneState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.2.2 MicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.2.3 MicButtonEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.2.4 MicrophoneSensitivityUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.2.5 MicStateChangedInRunningMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.3 Meeting Agenda</u></b>		
3.2.3.1 MeetingTitleChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
3.2.3.2 MeetingStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.3.3 MeetingAgendaChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.3.4 Agenda_ItemChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.3.5 keypad_login_code	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.2.4 Voting</b>		
3.2.4.1 VotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.4.2 VotingOutcome	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.4.3 GeneralVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.4.4 IndividualVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.4.5 VotingTemplateAdded	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.2.5 Timer</b>		
3.2.5.1 MeetingTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.2 MeetingTimerStateChange	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.3 DelegateTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.4 DelegateTimerStateChange	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.5 GroupTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.6 GroupTimerStateChange	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.7 SeatTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.8 SeatTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.9 ActiveAgendaItemTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.10 ActiveAgendaItemTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.11 VotingTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.5.12 VotingTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.2.6 Delegate</b>		
3.2.6.1 BadgeEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.2 DelegateOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.3 MeetingStartsUserInformation	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.4 DelegateUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.5 DelegateAdded	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.6 DelegateRemoved	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.6.7 DelegateSeatingUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
3.2.6.8 DelegatesInMeetingUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.7 Interactive</u></b>		
3.2.7.1 ScreenLockRemoved	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.7.2 DelegateScreenSet	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.7.3 CustomButtonEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.8 Audio</u></b>		
3.2.8.1 SetAudioSoftwareConfigurationById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.8.2 CurrentAudioSoftwareConfigurationUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.9 Logging</u></b>		
3.2.9.1 ConnectionLost_CentralUnit	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.9.2 ConnectionLost_Database	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.2.10 ButtonLED Event</u></b>		
3.2.10.1 ButtonPressed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.2.10.2 NextButtonPressed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.10.3 PriorButtonPressed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b><u>3.2.11 Interpretation</u></b>		
3.2.11.1 IMicrophoneStateUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.11.2 IActivatedChannelUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.11.3 SpeakSlowerRequest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.11.4 SpeakSlowerEnded	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.11.5 ActivePresetStateChanged	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b><u>3.2.12 Recording ( AE-R &amp; Plixus Core only)</u></b>		
3.2.12.1 WcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b><u>3.2.13 Wireless Coupling</u></b>		
3.2.13.1 WcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.13.2 WcapCoupledUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.2.13.3 AllWcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
<b><u>Client to server</u></b>		
<b><u>3.3.1 General</u></b>		
3.3.1.1 Connect	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.1.2 Subscribe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.1.3 Unsubscribe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.1.4 Notification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.1.5 ReboothSystem	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.3.2 Room</u></b>		
3.3.2.1 GetRoomServerName	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.2.2 GetActiveRoomInfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.3 GetStatusOfCuAndDbConnection	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.2.4 SetInitializationState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.5 GetAllUnitsState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.6 SetDefaultVolumeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.7 GetDefaultMicrophoneModeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.8 SetDefaultMicrophoneModeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.9 SetVolumeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.10 AddSynoptic	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.2.11 GetAllSeats	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.12 GetBooths	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.13 EditSeat	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.14 GetAllUnits	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.15 SetOperatingMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.16 GetOperatingMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.2.17 ClearAllMeetingAndDelegateData	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b><u>3.3.3 Microphone</u></b>		
3.3.3.1 SetState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.2 SetMicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.3 SetActivationType	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.4 SetMicrophoneSensitivity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.5 Get	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.6 GetState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



API Module	Cocon for Plixus Core	Plixus Core
3.3.3.7 ClearMicrophoneList	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.8 GetMicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.3.9 ActivateNextMicrophone	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>3.3.4 Meeting Agenda</b>		
3.3.4.1 SetMeetingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.2 StartEmptyMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.3 GetMeetingsForToday	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.4 GetAllMeetingsForDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.5 GetMeetingById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.6 SetActiveAgendaItemByDirection	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.7 SetActiveAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.8 GetAgendaItemInformationInRunningMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.9 GetAgendaItemInformationInMeetingById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.10 GetAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.11 StartMeetingViaMeetingTemplate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.12 StartMeetingViaXMLimport	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.13 AddAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.14 GetLoginMethodForMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.15 RemoveAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.16 DeleteAllAgendaItems	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.17 MoveAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.18 EditAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.19 AddDelegateToMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.20 AddDelegatesToMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.21 RemoveDelegateFromMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.22 GetDelegateSeating	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.23 AddDelegateToSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.24 AddDelegatesToSeats	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.25 RemoveDelegateFromSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.26 ApplyDelegatesToSystem	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.27 EndActiveMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.4.28 ResendMeetingData	<input checked="" type="checkbox"/>	<input type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
<b>3.3.5 Voting</b>		
3.3.5.1 SetVotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.5.2 AddInstantVote	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.5.3 AddVotingTemplate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.5.4 GetVotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.5.5 GetAmountOfVotes	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.5.6 GetGeneralVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.6 Timer</b>		
3.3.6.1 Get	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.2 GetDelegateTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.3 GetSeatTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.4 GetGroupTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.5 GetActiveTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.6 SetDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.7 SetGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.8 SetGlobalDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.9 SetGlobalSeatTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.10 SetGlobalGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.11 ResetDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.12 ResetGlobalDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.13 ResetGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.14 ResetGlobalGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.15 ChangeDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.6.16 ChangeGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.7 Delegate</b>		
3.3.7.1 GetDelegatesInMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.2 GetDelegatesInCurrentMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.3 ClearAllDelegatesAndGroups	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.4 GetAllDelegates	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.5 AddDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.6 RemoveDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.7.7 EditDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
3.3.7.8 ImportDelegatesFromFile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.8 Interactive</b>		
3.3.8.1 SetDelegateScreen	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.2 RemoveScreenLock	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.3 SetCustomButtonOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.4 SetCustomButtonOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.5 SetCustomLabelOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.6 SetCustomLabelOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.7 SetCustomScreenOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.8.8 SetCustomScreenOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.9 Audio</b>		
3.3.9.1 GetCurrentAudioSoftwareConfiguration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.9.2 GetCurrentAudioSoftwareConfigurationInDetails	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.9.3 GetAudioSoftwareConfigurations	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.9.4 GetAudioGroupsInCurrentConfiguration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.9.5 SetAudioSoftwareConfigurationById	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.9.6 SetAudioSoftwareConfigurationByName	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.9.7 ChangeMatrixNodeStatus	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.9.8 GetMatrixNodeStatus	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.9.9 SaveAudioSoftwareConfiguration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>3.3.10 Logging</b>		
3.3.10.1 GetCUState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.10.2 GetUnitErrorState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.11 ButtonLED Event</b>		
3.3.11.1 SetButtonsEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.11.2 SetLED	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.11.3 SetNextButtonEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.11.4 SetPriorButtonEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.11.5 SetNextButtonLedStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>

API Module	Cocon for Plixus Core	Plixus Core
3.3.11.6 SetPriorButtonLedStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.12 External</b>		
3.3.12.1 Send message to Operator	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.12.2 NotifyExternalCall	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.13 Interpretation</b>		
3.3.13.1 GetAllBoothsInRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.2 AddLanguage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.3 RemoveLanguage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.4 ClearLanguages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.5 AddInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.6 DeleteInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.7 ApplyInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.8 GetActiveInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.9 GetAllInterpreterPresets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.10 GetAllLanguages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.11 GetAllChannels	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.12 AddChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.13 EditChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.14 DeleteChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.15 RemoveAllChannels	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.16 SetDisplayText	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.17 SaveInterpreterPresetAs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.18 GetAllChannelRealtimeData	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.19 GetSpeakSlowerRequestList	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.13.20 GetInterpreterPresetById	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>3.3.14 API calls for Configurations</b>		
3.3.14.1 GetCulpAddress	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.14.2 SetCulpAddress	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.14.3 RestartRoomServer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3.3.14.4 GetCoconServerVersion	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>3.3.15 Recording ( AE-R &amp; Plixus Core only)</b>		

API Module	Cocon for Plixus Core	Plixus Core
3.3.15.1 GetRecordingFilesInfo	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.15.2 Startrecording	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.15.3 Stoprecording	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.15.4 Getrecordingstate	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>3.3.16 Wireless coupling</b>		
3.3.16.1 Getwcapinfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.16.2 Getallwcapinfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.3.16.3 SetWcapCoupledMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 2 General communication structure

### 2.1 Communication format

#### 2.1.1 REST/JSON

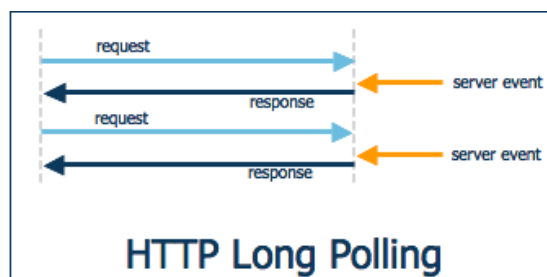
The technology being used for the CoCon API is REST (Representational State Transfer) with the JSON (JavaScript Object Notation) used. More information can be found at:

[http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)

<http://en.wikipedia.org/wiki/Json>

#### 2.1.2 Long polling

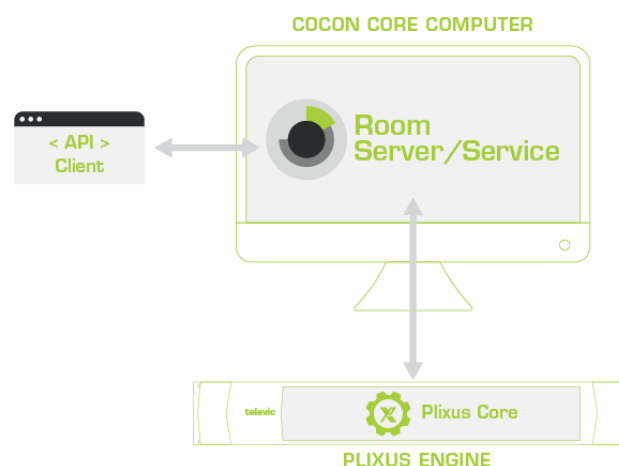
Since REST is a framework that does not natively allow to send messages from the server-side to the client-side, the CoCon API uses long polling to make sure that these messages can be sent. This means that a connection between client and server is kept open for a longer period of time. Over this connection, the server can then send information when and if it is available. See also the figure below:



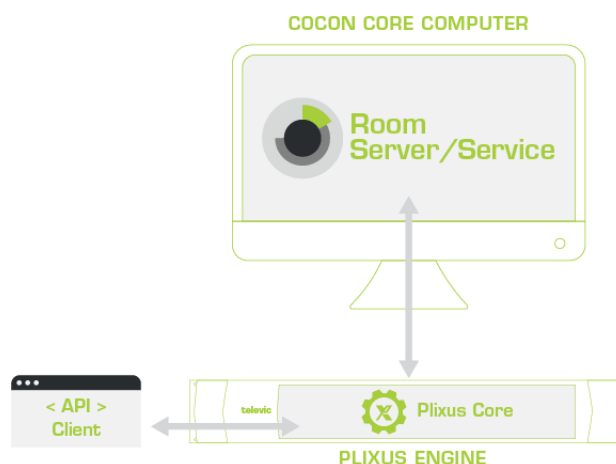
### 2.2 Communication methods

Different Plexus setups are available. Depending on the Plexus system you have installed different use cases of the API are possible:

- **CoCon for Plexus Core:** this is the CoCon software specifically designed to work with Plexus Core. Also in this case the API will connect to the room server.



- **Plixus Core:** in this system part of the conference logic resides in the engine (such as initialization and audio settings). Configuration of these features goes through a web server (which navigates to the IP address of the engine). In this case the API connects to the Plexus engine.



**Note:**

- The Plixus Core API is a subset of the Cocon Core API. Only a limited set of modules is possible with the Plixus Core API. At the beginning of this document you can find an exhaustive list of all API modules together with the system that supports it.

## 2.3 Publish/Subscribe

The CoCon API uses a Publish/Subscribe mechanism. This means that external clients who connect to the API can choose of which events they are being informed.

As can be seen below, there are commands contained in the API to subscribe or unsubscribe from each of the “Models”. Each model contains a certain set of functionality in relation to the conference room and software. The currently available models are:

- **Room:** This module publish all necessary events for Room Initialization, Volume change, unit state.
- **Microphone:** this includes all events relating to microphone state change (on, off, in request) and changes of the microphone mode.
- **Meeting\_Agenda:** this model contains all events with relation to the meeting (start, stop, pause, title change) and the agenda control in the meeting (next, previous, ...)
- **Voting:** this model will publish all information with regard to voting. This can be voting start, stop, pause, resume as well as individual and global voting results (if and when available).
- **Timer:** this model will publish all information about timers
- **Delegate:** this model contains all information about delegates in the meeting: detailed information, badge events,
- **Interactive:** this model allows controlling interactive meeting sessions. This is only applicable for the Televic uniCOS (PlixUs) CU and system.
- **Audio :** N/A
- **Logging:** This model will publish all status of connection between CoCon server and Database, Broken unit status and CU status.
- **ButtonLED\_Event:** this model allows to listen to button events on the delegate units and set LEDs accordingly.

Some remarks specifically for this model:

- The order to use the commands is typically the following:
  - “SetButtonsEnabled” to enable one or more buttons (section 3.3.11.1).
  - Listen to the events “ButtonPressed”, which mean that a button was pressed on a unit (section 3.2.10.1).
  - Control the LEDs on the unit using the command “SetLED” (section 0).
- **Note that CoCon does not guarantee that button events will always be sent out or set LEDs always responded upon. This is because in certain phases of operation (most notably voting sessions), the buttons and LEDs are used by CoCon for voting functionality.**

- When a voting session is active, the buttons and LEDs are controlled by CoCon, and cannot be controlled externally by the API. This means that the ButtonLED\_Event commands and events can only be used or listened to if the VotingState is equal to "Clear". See section **Error! Reference source not found.** For the notification of this state to the API client.
- When the voting session is ended, the buttons that were enabled before the voting session, by the ButtonLED\_Event commands, are again enabled. However, the LED state is not restored. Keeping track of LED-state and re-setting this state is the responsibility of the API client.
- The functionality of this model depends heavily on the type of CU (Central Unit) and/or DU (Delegate Unit) to which CoCon is connected.
- **External:** This module send event for special button pressed by chairmen units.
- **Interpretation:** This module sent all the necessary events related to Interpretation.
- **Intercom:** This module send the events related to Intercom communication.

#### Note the following:

- Upon connection, clients are by default subscribed to almost ALL models. If some information is not useful for the client in question, he can send commands to unsubscribe from certain of these models (see below for the Unsubscribe command).
- Clients can send commands to the CoCon API, independently from the models to which they are described. This means for example: a client who is NOT subscribed to the Microphone model CAN send a command to turn on a microphone, but will NOT be notified that the microphone has been turned.  
This means that the API client needs to judiciously select the models to which it will subscribe and unsubscribe.
- Notification api call time-out :
  - Response type 408 (intended to indicate time-outs)
  - Time-out time of 30sec
  - Response message : "Notification request timeout"
- Invalid notification (invalid id or id which has expired) :
  - Response type 400 (bad request) for expired or invalid id
  - Make client id expired if there is no request of get notification in 180sec.
  - Response message
    - in case of expired session: "{"Notification" : "Error 400"}"
    - in case of invalid id: "{"Notification" : "Error 400"}"

## 2.4 Setting up and operating the CoCon API connection

### 2.4.1 General

The CoCon API connection consists of a number of URI's (Uniform Resource Identifier) which are exchanged between the API client and the CoCon Server over a TCP/IP-connection.

In this document, it is assumed that the API Client and the CoCon Room Server are running on the same computer. This means that the URI's are using "localhost" as the target machine of the API-calls. Of course, if the CoCon Room Server is running on another machine and you are connecting to this machine, then you have to exchange "localhost" to the IP address or host name of the target machine.

Basically, the CoCon API uses two (or more) parallel HTTP-connections. These are the following:

- A "notification" connection that the client and Server keep open in case any information need to be transferred from the Server to the client.
- Any other connection that is used to transfer information from the client to the Server.

The notification connection should be initialized by the client, and is kept open by both sides. Then one of two things can happen:



- The Server has information to send to the client and uses the notification connection for this. The information is transferred and the connection is closed (as per the HTTP standard). After this, the client **needs to create a new notification connection**. The Server will guarantee that no information is lost during this short period where the client has no connection to the Server.  
Note that the new notification connection needs to be created within 60 seconds. If this does not happen, the Server will discard the client data.
- The HTTP-connection times out before the Server has any information to send to the client. At this point, the client **needs to create a new notification connection**. Again, the Server will guarantee that no information is lost.

## 2.4.2 REST/JSON Example

This section contains an example of the creation and operation of the API connection to the CoCon Server using the REST/JSON format.

The first URI that has to be sent to the CoCon API is the Connect URI:

<http://localhost:8890/CoCon/Connect>

As a reply, the following JSON string will be sent back:

```
“{“Connect”:true,“id”:"3f5b8afe-6efb-4428-8762-e39d42a0762d"}”
```

The first parameter indicates that the connection has succeeded; the second parameter is the Id of this connection that has to be used when opening the notification connection. In order to open this notification connection, the following URI has to be sent to the Server:

<http://localhost:8890/CoCon/Notification/id=3f5b8afe-6efb-4428-8762-e39d42a0762d>

Now the notification connection is open and the Server can send updates to the client. This can for example be the following:

```
{“MicrophoneMode”: {“Mode” :”Operator” ; “MaxNrActive” : <MaxActiveMicNumber>;  
“MaxNrRequest”: <MaxRequestMicNumber> }}
```

This is an update of the Microphone Mode that is currently being used in the conference software and room. As discussed above, the CoCon Server will send all applicable updates to the API clients that are subscribed to the model that the event falls under. By default, clients are subscribed to all models. Unsubscribing can be done by sending the following URI over the API:

<http://localhost:8890/CoCon/Unsubscribe/?Model=Logging&id=3f5b8afe-6efb-4428-8762-e39d42a0762d>

This URI unsubscribes the current client from the Logging model. From this point onwards, no more events from the specified model will be sent to this API client.

When a microphone is activated, the following information will be sent over the notification connection:

```
{“MicrophoneState”:{“Speakers”:[1,5];“Requests”:[7]}}
```

...which indicates that the microphones on seats 1 and 5 are on, and that the microphone on seat 7 is in request. After receiving this information, the client must re-open the notification connection.

The other way around, if the API client wants to activate a microphone, the following URI must be sent to the CoCon Server:

<http://localhost:8890/CoCon/Microphone/SetState/?State=On&SeatNr=9>

...to turn on the microphone on seat 9.

### 2.4.3 Server return values

Over-all feedback from the CoCon during the 3<sup>rd</sup> party API communication.

**Command request based on integer data type:**

>> 0: Command received successfully to CoCon server and handle.

>> 1: Command sends from 3<sup>rd</sup> party with in-correct parameter value / parameter list.

>> -1: Command received correctly on CoCon server but exception happen due to the abnormal value send by user.

**Command request based on the string data type:**

Server returns data when command successful handled.

Server returns hint information for the unhandled command.

## 3 Operational commands

This chapter contains the description of all commands making up the API. Note that the format has been kept general.

### 3.1 Data models

These models will be used to exchange information from and to the CoCon API.

#### 3.1.1.1 AuthorityAssigned

This represents authority present voted, and register in the voting

Properties:

Name	Description	Type
Present	The number of authorities that present in the voting	int
Voted	The number of authorities that voted in the voting	int
Register	The number of authorities that Register in the voting	int

#### 3.1.1.2 Meeting

Data model that represents meeting object

Properties:

Name	Description	Type
Id	The id of the meeting	int
Title	The title of the meeting	string
Description	The description of the meeting	string
StartTime	The start time of the meeting	string
State	The state of the meeting Paused; Running; Ended	string

#### 3.1.1.3 VotingCountWeight

Data model that represents the voting count and weight pair

Name	Description	Type
Count	The count of the voting	int
Weight	The weight of the voting	double

#### 3.1.1.4 VotingOption

Data model that represents the voting option

Name	Description	Type
Id	The id of the voting option	int

Name	The name of the option	string
Color	The color of the voting option	string

### 3.1.1.5 VotingOptionVoteDetail

Data model that represents the votingCountWeight of the voting option

Name	Description	Type
Option	The related voting option	<a href="#">VotingOption</a>
OptionVoted	The voted count and weight of the option	<a href="#">VotingCountWeight</a>

### 3.1.1.6 VotingResultSummary

Data model that represents the summary result of the voting

Name	Description	Type
Total	The total voting count and weight	<a href="#">VotingCountWeight</a>
Voted	The voted voting count and weight	<a href="#">VotingCountWeight</a>
NotVoted	The not voted voting count and weight	<a href="#">VotingCountWeight</a>
Options	Each voting option count and weight	<a href="#">VotingOptionVoteDetail</a>
AuthorityAssigned	The authority used in the voting	<a href="#">AuthorityAssigned</a>

### 3.1.1.7 IndividualVote

Data model that represents the summary result of the voting

Name	Description	Type
DelegateID	The ID of the delegate that has cast the vote	int
VotingOptionID	The ID of the voting option for which the vote was cast	int

### 3.1.1.8 Timer

Data model that represents the timer

Name	Description	Type
TotalTime	The total time (hh:mm:ss)	string
TimeUsed	Used time (hh:mm:ss)	string
CountingDown	Show counting down or up in CoCon	bool
WarningTime	Warning time in CoCon (hh:mm:ss)	string
enabled	Set enabled or disabled timer	bool

### 3.1.1.9 BaseObjectTimer

Data model that represents the object's timer

Name	Description	Type
Timer	The timer of the object	<a href="#">Timer</a>

### 3.1.1.10 DelegateTimer : BaseObjectTimer

Data model that represents the delegate's timer

Name	Description	Type
DelegateId	The delegate's Id	int

### 3.1.1.11 GroupTimer : BaseObjectTimer

Data model that represents the group's timer

Name	Description	Type
GroupId	The delegate's Id	int

### 3.1.1.12 SeatTimer : BaseObjectTimer

Data model that represents the timer of the seat

Name	Description	Type
SeatNr	The number of the seat	int

### 3.1.1.13 Delegate

Data model that represents the delegate

Name	Description	Type
Id	The id of the delegate	int
FirstName	The first name of the delegate	string
Name	The name of the delegate	string
Street	The street that the delegate lives	string
StreetNumber	The street number of that street	int
PostCode	The post code of the delegate that he lives	string
City	The city that the delegate lives	string
Country	The country that the delegate lives	string
Title	The title of the delegate	string
BirthDate	The birth date of the delegate	string (format yyyy-MM-dd)
District	The district of the delegate lives	string
Biography	The delegate's biography	string
Groups	The groups of the delegate	An array of <a href="#">Group</a>

### 3.1.1.14 Group

Data model that represents the group

Name	Description	Type
Id	The id of the group	int
Name	The name of the group	string

### 3.1.1.15 Agendaltem

Data model that represents the agenda item

Name	Description	Type
Id	The id of the Agenda item	string
Title	The title of the agenda item	string
Description	The description of the agenda item	string
Type	The type of the agenda item VotingAgendaltem LecturerAgendaltem Agendaltem	string
State	The state of the agenda item	string
Children	The Children agenda item of this agenda item	List< <a href="#">Agendaltem</a> >

### 3.1.1.16 VotingAgendaltem:Agendaltem

Data model that represent the voting agenda item

Name	Description	Type
VotingOptions	The voting options	List< <a href="#">VotingOption</a> >

### 3.1.1.17 LecturerAgendaltem:Agendaltem

Data model that represent the lecturer agenda item

Name	Description	Type
Lectures	The lecturers of this item	List< <a href="#">Delegate</a> >

### 3.1.1.18 AudioGroup

Data model that represent Audio group which contains routing input group and routing output group.

Name	Description	Type
Id	The Id of the audio group	int
Type	The type of the audio group(input/output)	string
Name	The name of the audio group	string

### 3.1.1.19 AudioMatrixNode

Data model that represent Audio matrix node.

Name	Description	Type
------	-------------	------

InputGroupName	The name of the routing input group	string
OutputGroupName	The name of the routing output group	string
MixValue	The mix value of the matrix node(-100~0)	int
Status	The status of the matrix node(enable or not)	bool
BackupStatus	The backup status of the matrix node(enable or not)	bool

### 3.1.1.20 AudioSoftwareConfiguration

Data model that represent Audio software configuration basic information.

Name	Description	Type
Id	The Id of the audio software configuration	int
Name	The name of the audio software configuration	string

### 3.1.1.21 AudioSoftwareConfigurationDetailed

Data model that represent Audio software configuration detail information.

Name	Description	Type
Id	The Id of the audio software configuration	int
Name	The name of the audio software configuration	string
AudioMatrixNodes	All the audio matrix nodes	List< <a href="#">AudioMatrixNode</a> >
AudioGroups	All the audio groups	List< <a href="#">AudioGroup</a> >

### 3.1.1.22 ScreenControlOptions

Properties: None (the inheriting objects will specify the necessary properties).

### 3.1.1.23 HomeControl : ScreenControlOptions

Properties: None

### 3.1.1.24 DiscussionControl : ScreenControlOptions

Properties:

Name	Description	Type
video	Set video viewing mode 0: disabled; no video shown 1: normal 2: full screen	int

### 3.1.1.25 AgendaControl : ScreenControlOptions

Properties:

Name	Description	Type
select	Select and show active agenda item	bool

### 3.1.1.26 DelegatesControl : ScreenControlOptions

Properties: None

### 3.1.1.27 VotingControl : ScreenControlOptions

Properties:

Name	Description	Type
res	Show voting results	bool
graph	Show voting graph	bool

### 3.1.1.28 DocumentsControl : ScreenControlOptions

Properties:

Name	Description	Type
doc	Id of the document to select	string
lck	Lock the document (no selection)	bool
page	Page number to be scrolled to	int
docfs	Show the document fullscreen	bool

### 3.1.1.29 MessagesControl : ScreenControlOptions

Properties: None

### 3.1.1.30 ServicesControl : ScreenControlOptions

Properties: None

### 3.1.1.31 ConfigurationControl : ScreenControlOptions

Properties: None

### 3.1.1.32 Unit

Data model that represent unit(like: microphone unit, vote unit,etc)

Name	Description	Type
RoomId	The Id of the room	int
SystemUnitId	The Id of the unit	int
SerialNumber	The serial number of the unit	string
UnitCapabilitiesInfo	The capabilities information of the unit	string
WirelessCapability	If the unit has the wireless capability	bool
VersionInfo	The version of the unit	string

### 3.1.1.33 Channel



Name	Description	Type
Id	The Id of the preset	int
Order	The order of the channel	int
Language	The language name	string
AudioRouting	Set audio routing for the channel. Plixus Core & CoCon For Plixus Core: 'comma-separated-booth-numbers', Disabled, External, EmptyBooth	string
DisplayText	The text will be displayed	string
AutoFloor	The channel is auto floor	bool
BackupChannel	The backup channel name	String
IsRecording	Channel recording or not. Optional parameter.	bool

### 3.1.1.34 InterpreterPreset

Name	Description	Type
Id	The Id of the interpreter preset	int
Name	The name of the interpreter preset	string
Description	The description of the interpreter preset	string
DeskMode	The mode between desks	int
BoothMode	The mode between booths	int
State	The state of the interpreter preset	string

### 3.1.1.35 DeskUnitInChannel

Name	Description	Type
Name	The name of the desk unit	string
ActiveChannel	The index of the active channel	int

### 3.1.1.36 ChannelRealtimeData

Name	Description	Type
Order	The order of the channel	int
Language	The name of the language	string
Booth	The name of the booth	string
Desks	The list of the desks in the channel	List< <a href="#">DeskUnitInChannel</a> >
Relay	The relay of the channel	string
Quality	The quality of the channel	int

## 3.2 Server to client

This section describes all commands that the CoCon Server will send to the API clients, notifying them of any events that have happened in the conference software or room.

As discussed above, the functionality is divided into models. Each model will be discussed in a separate section below. The first section is a general section.

### 3.2.1 Room

#### 3.2.1.1 InitializationState

API Module	Cocon for Plixus Core	Plixus Core
InitialisationState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This command is sent when Initialization state changes. The initialization states are:

- Start
- Stop
- Retrieve
- Auto
- Reset

**Name:** InitialisationState

**REST/JON Example:**

```
{"InitialisationState":"AUTO"}
```

(Note: uppercase letters will be used on return)

#### 3.2.1.2 UnitStateChanged

API Module	Cocon for Plixus Core	Plixus Core
UnitStateChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This notification is sent when a unit's online state changes, i.e. if it goes offline or comes online again.

**Parameters:**

Name	Description	Type
UnitId	The id of the unit	int
State	The new state of the unit 0: offline 1: online	<0/1>

**Example:**

**Wired unit's notification :**

```
{"UnitStateChanged":{"Id":9,"State":0}}
```

**Wireless unit's notification :**

```
{"UnitStateChanged":{"Id":6,"State":0,"BatteryState":1448,"Snr":0,"PacketLoss":3}}
```

#### 3.2.1.3 VolumeChangedForRoom

API Module	Cocon for Plixus Core	Plixus Core
VolumeChangedForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This notification will send once default volume changed in room.

**Parameters:**

Name	Description	Type
Id	Connected room id	int
Volume	Volume of the room	int

**REST/JON Example:**

```
{"VolumeChangedForRoom":{"Id":1,"volume":24}}
```

**3.2.1.4 UnitError**

API Module	Cocon for Plixus Core	Plixus Core
UnitError	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This notification will send out once the unit erro happen.

**Parameters:**

Name	Description	Type
State	State of the unit	string
UnitId	UnitId	int

**REST/JON Example:**

```
{"UnitError":{"State":"Error","UnitId":6}}
```

**3.2.2 Microphone**

**3.2.2.1 MicrophoneState**

API Module	Cocon for Plixus Core	Plixus Core
MicrophoneState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This command is sent every time the state of a microphone changes. The different options of a microphone state are:

- On = Speakers
- In Request = Requests
- Off

As indicated, the seat-numbers of the microphones that are On are contained in the “Speakers” array. The seat-numbers of the microphones that are In Request are contained in the second parameter.

**Name:** MicrophoneState

**Parameters:**

Name	Description	Type
------	-------------	------

Speakers	An array of the seatNrs of all currently active microphones	Array of integers
Requests	An array of the seatNrs of all the microphones that are currently in request	Array of integers
Replies	An array of the seatNrs of all the microphones that are currently in replies	Array of integers

**REST/JON Example:**

```
{"MicrophoneState":{"Speakers":[6],"Requests":[],"Replies":[]}}
```

### 3.2.2.2 MicrophoneMode

API Module	Cocon for Plixus Core	Plixus Core
MicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This command is sent every time the microphone mode changes.

**Name:** MicrophoneMode

**Parameters:**

Name	Description	Type
Mode	The Microphone mode. Values: Operator DirectSpeak Request Vox OnlyRequest	string
MaxNrActive	The maximum number of microphones active at the same time	int
MaxNrRequest	The maximum number of microphones in request at the same time	int
AllowRequest	bool indicating whether the delegates are allowed to request. Not applicable for all modes. Optional	bool
AllowCancelRequest	bool indicating whether the delegates are allowed to cancel their request. Not applicable for all modes. Optional	bool
FIFO	bool indicating whether the FIFO mode is active Not applicable for all modes. Optional	bool
AllowSwitchOffMic	Bool indicating whether the delegate are allowed to switch off their mic or not.	bool
ReplyRequest	Indicating that the ReplyRequest (Replik) function is enabled or not on Server.	bool

**REST/JON Example:**

```
{"MicrophoneMode":{"Mode":"OnlyRequest","MaxNrActive":4,"MaxNrRequest":200,"AllowRequest":true,"AllowCancelRequest":false,"FIFO":false,"ActivationType":"Toggle","AllowSwitchOffMic":True,"ReplyRequest":false}}
```

### 3.2.2.3 MicButtonEvent

API Module	Cocon for Plixus Core	Plixus Core
MicButtonEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is sent every time a microphone button is pushed. It contains information about the seat that has generated the event.

**Name:** MicButtonEvent

**Parameters:**

Name	Description	Type
SeatNr	The number of the seat	int
Event	The button event (up or down)	string

**REST/JON Example:**

```
{"MicButtonEvent":{"SeatNr":2,"Event":"up"}}
```

### 3.2.2.4 MicrophoneSensitivityUpdated

API Module	Cocon for Plixus Core	Plixus Core
MicrophoneSensitivityUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event notification for the seat (Microphone) sensitivity changed or updated.

**Name:** MicrophoneSensitivityUpdated

**Parameters:**

Name	Description	Type
SeatId	The number of the seat	int
Sensitivity	Value of the sensitivity Range of value +12 to -12 Ex. (12,11,10,9...-10,-11,-12)	int

**REST/JON Example:**

```
{"MicrophoneSensitivityUpdated":{"SeatId":1,"Sensitivity":-6}}
```

### 3.2.2.5 MicStateChangedInRunningMeeting

API Module	Cocon for Plixus Core	Plixus Core
MicStateChangedInRunningMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event notification for the delegate on seat (Microphone) updated.

**Name:** MicStateChangedInRunningMeeting

**REST/JON Example:**

```
{"MicStateChangedInRunningMeeting":{"SeatNr":1,"State":"On","User":{"Id":463,"FirstName":"Marcel","Name":"Ackermann","MiddleName":null,"Street":"Route Principale 11c","StreetNumber":0,"PostCode":"2803","City":"Bourrignon","Country":"","Title":"Député","BirthDate":"","District":"","Biography":"","Groups":[{"Id":21,"Name":"PDC"}],"SeatNumber":1,"BadgeNumber":"14EFB4E2","FingerprintData":"","KeypadLoginCode":""}}
```

### 3.2.3 Meeting\_Agenda

#### 3.2.3.1 MeetingTitleChanged

API Module	Cocon for Plixus Core	Plixus Core
MeetingTitleChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This command is sent every time the active meeting title changed.

**Name:** MeetingTitleChanged

**Parameters:**

Name	Description	Type
MeetingId	ID of the meeting	int
NewTitle	The new meeting title	string

**REST/JON Example:**

```
{"MeetingTitleChanged":{"Title":"2/11/2014 new Meeting"}}
```

#### 3.2.3.2 MeetingStatus

API Module	Cocon for Plixus Core	Plixus Core
MeetingStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This command is sent every time the state of current active meeting changed.

**Name:** MeetingStatus

**Parameters:**

Name	Description	Type
MeetingId	The id of the meeting	int
State	The state of the meeting include: Paused; Running; Ended	string

**REST/JSON Example:**

```

{"MeetingStatus":{"MeetingId":14,"State":"Paused"}}
{"MeetingStatus":{"MeetingId":14,"State":"Running"}}
{"MeetingStatus":{"MeetingId":14,"State":"Ended"}}

```

### 3.2.3.3 MeetingAgendaChanged

API Module	Cocon for Plixus Core	Plixus Core
MeetingAgendaChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This notification is sent out when the agenda has changed, i.e. when an agenda item is added, removed, edited or moved. It is NOT sent when the active agenda item changes, since there is already another API command for that (Agenda\_ItemChanged).

The API client that is interested in the changes in the agenda, can get the entire agenda with GetAgendaItemInformationInRunningMeeting

#### Parameters:

Name	Description	Type
MeetingId	The id of the meeting	int

#### REST/JSON Example:

```

{"MeetingAgendaChanged":{"MeetingId":9}}

```

### 3.2.3.4 Agenda\_ItemChanged

API Module	Cocon for Plixus Core	Plixus Core
Agenda_ItemChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This command is sent every time the state of an agenda item changes.

**Name:** Agenda\_ItemChanged

#### Parameters:

Name	Description	Type
MeetingId	The id of the meeting	int
Agenda_ItemChanged	The changed agenda item	AgendaItem
IdInDb	In CoCon DB Agenda ID	int

#### REST/JON Example:

```

{"MeetingId":4,"Agenda_ItemChanged":{"Id":1,"Title":"","Description":"","Type":"AgendaItem","State":"active","Children":[]}}

```

Ex:2

```

{"MeetingId":74,"Agenda_ItemChanged":{"Id":4,"Title":"","Description":"","Type":"VotingAgendaItem","State":"ended","VotingOptions":[{"Id":397,"Name":"1","Color":"#FF006400"}, {"Id":398,"Name":"2","Color":"#FF008000"}, {"Id":399,"Name":"3","Color":"#FFFFFF00"}, {"Id":400,"Name":"4","Color":"#FFFF0000"}, {"Id":401,"Name":"5","Color":"#FF8B0000"}],"Children":[],"IdInDb":131}}

```

### 3.2.3.5 Keypad\_login\_code

API Module	Cocon for Plixus Core	Plixus Core
keypad_login_code	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This command is sent from Plixus FMM (IDC) to CoCon whenever user input numeric code and press the Ok.

**Name:** keypad\_login\_code

**Parameters:**

Name	Description	Type
SeatNr	Seat number	int
LoginCode	Code entered by user from Plixus FMM (IDC)	string

**REST/JON Example:**

```
{"keypad_login_code":{"SeatNr":4,"LoginCode":"1234"}}
```

## 3.2.4 Voting

### 3.2.4.1 VotingState

API Module	Cocon for Plixus Core	Plixus Core
VotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The voting state of current active voting agenda item

**Name:** VotingState

**Parameters:**

Name	Description	Type
Id	ID of the VotingAgendaItem	int
State	The state of voting, include: Start; Stop Pause Clear	string
VotingTemplate	The voting template that voting agenda item based on	string

**REST/JON Example:**

```
{"VotingState":{"Id":11,"State":"Start","VotingTemplate":"3_Vote_Public"}}
```

### 3.2.4.2 VotingOutcome

API Module	Cocon for Plixus Core	Plixus Core
VotingOutcome	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The outcome message of voting result



**Name:** VotingOutcome

**Parameters:**

Name	Description	Type
Id	ID of the VotingAgendaItem	int
VotingOptionId	ID of the VotingOption that won, if applicable 0 if not applicable	int
Outcome	The outcome message of voting result	string

**REST/JON Example:**

```
{"VotingOutcome":{"Id":11,"VotingOptionId":1147,"Outcome":"Pass option1"}}
```

### 3.2.4.3 GeneralVotingResults

API Module	Cocon for Plixus Core	Plixus Core
GeneralVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The general voting results of current voting

**Name:** GeneralVotingResults

**Parameters:**

Name	Description	Type
Id	ID of the VotingAgendaItem	int
VotingResults	The general results of current voting	<a href="#">VotingResultSummary</a>

**REST/JON Example:**

```
{"GeneralVotingResults":{"Id":11,"VotingResults":{"Total":{"Count":10,"Weight":10},"Voted":{"Count":0,"Weight":0},"NotVoted":{"Count":10,"Weight":10},"AuthorityAssigned":{"Present":0,"Voted":0,"Register":0}}}}
```

Ex 2.

```
{"GeneralVotingResults":{"Id":5,"VotingResults":{"Total":{"Count":2,"Weight":2.0},"Voted":{"Count":2,"Weight":2.0},"NotVoted":{"Count":0,"Weight":0.0},"Options":[{"Id":31,"Name":"+","Color":"#FF008000","Votes":{"Count":1,"Weight":1.0},"OptionChoice":1},{"Id":32,"Name":"0","Color":"#FFFFFF00","Votes":{"Count":1,"Weight":1.0},"OptionChoice":2},{"Id":33,"Name":"-","Color":"#FFFF0000","Votes":{"Count":0,"Weight":0.0},"OptionChoice":3}],"AuthorityAssigned":{"Present":0,"Voted":0,"Register":0}}}}
```

### 3.2.4.4 IndividualVotingResults

API Module	Cocon for Plixus Core	Plixus Core
IndividualVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The individual voting results of current voting, whenever a new vote is available. Note the following:

- For every one vote, this event will only be sent once, at the moment of casting the vote.
- The delegate info and voting option info is not present in this command to optimize performance. Only IDs are available.

**Name:** IndividualVotingResults

**Parameters:**

Name	Description	Type
Id	ID of the VotingAgendaItem	int
VotingResults	An array with all the individual votes	Array of <a href="#">IndividualVote</a>

REST/JON Example:

```
{"IndividualVotingResults":{"Id":5,"VotingResults":[{"DelegatId":310,"VotingOptionId":32}]}}
```

**3.2.4.5 VotingTemplateAdded**

API Module	Cocon for Plixus Core	Plixus Core
VotingTemplateAdded	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Event notification for the added new voting template or update voting template.

**Parameters:**

Name	Description	Type
Title	Name of the Voting template and will keep the title due that agenda	string
NrOfOptions	Number of voting options	int
IsPrinted	Enabled the function for print after voting stop.	bool
IsVotingNumber	Enabled Symbol or Number for voting choice <b>Note: not applicable after 5.0 cocon</b>	bool
AbstainOption	Choose the Voting abstain Options Value :1 to 3	int
AbstainIndex	Select abstain index for move voting for Value : 1 to .. based on the voting options.	int
Description1	Description for voting option 1	string
Option1 (Option Color)	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. <b>Note: Make sure for correct RGB value.</b>	string
SymbolIndex1	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour1	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain1	Enabled the abstain function	bool
MsgOfMajority1	Message for majority	string
Description2	Description for voting option 2	string
Option2	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. <b>Note: Make sure for correct RGB value.</b>	string
SymbolIndex2	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour2	Voting LED color We support only 7 color now.	string

	Blue, White, Green, Yellow, Red, Magenta, Cyan	
IsAbstain2	Enabled the abstain function	bool
MsgOfMajority2	Message for majority	string
Description3	Description for voting option 3	string
Option3	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex3	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour3	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain3	Enabled the abstain function	bool
MsgOfMajority3	Message for majority	string
Description4	Description for voting option 4	string
Option4	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex4	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour4	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain4	Enabled the abstain function	bool
MsgOfMajority4	Message for majority	string
Description5	Description for voting option 5	string
Option5	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex5	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour5	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain5	Enabled the abstain function	bool
MsgOfMajority5	Message for majority	string
Description6	Description for voting option 6	string
Option6	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex6	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour6	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string

	Note: LED color not support for more than 5 button options.	
IsAbstain6	Enabled the abstain function	bool
MsgOfMajority6	Message for majority	string
Description7	Description for voting option 7	string
Option7	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex7	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour7	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan Note: LED color not support for more than 5 button options.	string
IsAbstain7	Enabled the abstain function	bool
MsgOfMajority7	Message for majority	string
Description8	Description for voting option 8	string
Option8	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex8	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour8	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan Note: LED color not support for more than 5 button options.	string
IsAbstain8	Enabled the abstain function	bool
MsgOfMajority8	Message for majority	string
Description9	Description for voting option 9	string
Option9	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex9	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour9	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan Note: LED color not support for more than 5 button options.	string
IsAbstain9	Enabled the abstain function	bool
MsgOfMajority9	Message for majority	string
Description10	Description for voting option 10	string
Option10	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex10	Buttons enabled for voting Index : 0 to 9 based on the options.	int

	Ex.	
LedColour10	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain10	Enabled the abstain function	bool
MsgOfMajority10	Message for majority	string
BadgeOption	Choose option for "Who can participate in voting" Value : 1 to 4 Ex. 4 = All voting units Note: badge options will be available based on the meeting type	int
OverallOption	Choose overall option "Who sees the overall results during the vote ?" Value : 1 to 4 1= no one	int
IndividualOption	Choose the option for "Who sees the individual voting results (during & after the vote)" Value: 1 to 5 1= No one ;Never	int
CanCorrect	Enabled the correction option Value : True means user can change voting	bool
HasPin	Enable the option for "Enter PinId for voting" Note: it will be used for IDC only.	bool
IsWeightUsed	Enable option to check the voting weight for voting result calculation.	bool
IsOperatorIndicated	Vote indication Enabled the Operator voting indication <b>Note: it can configure based on the individual voting setting.</b>	bool
IsSignageIndicated	Vote indication Enabled the Signage voting indication <b>Note: it can configure based on the individual voting setting.</b>	bool
PanelIndicationOption	Vote button indication Chose the vote panel indication option Value : 0 to 2 2 means "Real vote choice" <b>Note: it can configure based on the individual voting setting.</b>	int
EnableAgendaTimer	Enabled the timer for Agenda	bool
DurationOfAgendaTimer	Total Time for Agenda Ex. 00:05:00	String
WarningOfAgendaTimer	Warning time for the Agenda Ex. 00:01:00	String
CountDownOfAgendaTimer	Choose the agenda timer count-up / count-down	Bool
ExceedOfAgendaTimer	Allow exceed agenda timer	Bool
EnableVotingTimer	Enabled the timer for voting	Bool
DurationOfVotingTimer	Total Time for Agenda Ex. 00:05:00	String
WarningOfVotingTimer	Warning time for the Agenda Ex. 00:01:00	String
CountDownOfVotingTimer	Choose the agenda timer count-up / count-down	Bool
ExceedOfVotingTimer	Allow exceed voting timer	Bool
EnableQuorum	Enabled the Quorum Functionality	Bool
AbsoluteOfQuorum	Choose Quorum define function for Absolute or relative Ex. True means Absolute enabled False means Relative option selected. (enabled)	Bool

ValueOfQuorum	Value of the Quorum	decimal
OptionOfQuorum	Choose the options for Presence of determination. Value : 1 to 5	int
EnableQuorumMsg	Enabled the Quorum message option	bool
QuorumMsg	Message for Quorum	string
AbsoluteOfOutcome	Choose Majority define function for Absolute or relative Ex. True means Absolute enabled False means Relative option selected (enabled).	bool
ValueOfMajority	Value of the Majority	decimal
MajorityOption	Choose the majority options Value :1 to 2	int
EnableOutcomMsg	Enabled Outcome message option.	bool
PassedMsg	Message for the Pass Majority	string
FailedMsg	Message for the Fail Majority	string
HungMsg	Message for the Hung Majority	string
EnabledMajorityMsg	Enabled Majority Achieved Details	bool
HungVoteSetting	Choose option for the hung vote solution Value : 1 to 5 Ex. 0 = Stop voting automatically. 1 = Keep the vote active until the Operator or Chairman manually stops it. 2 = Restart voting with all options available 3 = Determined by chairmen votes in this voting session. 4 = Stop voting and the operator decides. 5 = Determined by chairmen votes in a new chairmen-only voting session.	Int
IsUnanimous	If value is true, means Voting options for the voting session decision made by operator even though user can choose his own chose as well. (this option will used by the Cocon Op app only now) Note: default value is false	bool

#### REST/JON Example:

```
{
  "VotingTemplateAdded": {
    "Id": 36,
    "Title": "TEST 501",
    "NrOfOptions": 5,
    "IsPrinted": false,
    "IsVotingNumber": false,
    "BadgeOption": 1,
    "OverallOption": 1,
    "Options": [
      {
        "Id": 157,
        "Description": "Pass",
        "Option": "#0000FF",
        "SymbolIndex": 0,
        "LedColour": "Red",
        "IsAbstain": true,
        "MsgOfMajority": ""
      },
      {
        "Id": 158,
        "Description": "ABS",
        "Option": "#FFFF0000",
        "SymbolIndex": 1,
        "LedColour": "Red",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Accepted"
      },
      {
        "Id": 159,
        "Description": "Fail",
        "Option": "#FFFFFF00",
        "SymbolIndex": 2,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Postponed"
      },
      {
        "Id": 160,
        "Description": "ISA",
        "Option": "#39FF14",
        "SymbolIndex": 3,
        "LedColour": "Yellow",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Rejected"
      },
      {
        "Id": 161,
        "Description": "XYZ",
        "Option": "#FF00FF",
        "SymbolIndex": 4,
        "LedColour": "Magenta",
        "IsAbstain": false,
        "MsgOfMajority": ""
      }
    ],
    "IndividualOption": 1,
    "CanCorrect": true,
    "HasPin": false,
    "IsWeightUsed": true,
    "IsOperatorIndicated": true,
    "IsSignageIndicated": true,
    "PanelIndicationOption": 1,
    "AgendaTimer": {
      "Id": 73,
      "EnableAgendaTimer": true,
      "DurationOfAgendaTimer": "00:05:00",
      "WarningOfAgendaTimer": "00:03:00",
      "CountDownOfAgendaTimer": false,
      "ExceedOfAgendaTimer": true
    },
    "VotingTimer": {
      "Id": 74,
      "EnableVotingTimer": true,
      "DurationOfVotingTimer": "00:05:00",
      "WarningOfVotingTimer": "00:05:00",
      "CountDownOfVotingTimer": false,
      "ExceedOfVotingTimer": true
    },
    "EnableQuorum": true,
    "AbsoluteOfQuorum": false,
    "ValueOfQuorum": 0.0,
    "OptionOfQuorum": 1,
    "EnableQuorumMsg": true,
    "QuorumMsg": ""
  }
}
```

works", "AbsoluteOfOutcome":false, "ValueOfMajority":10.0, "MajorityOption":1, "EnableOutcomMsg":true, "PassedMsg":"Majority achieved", "FailedMsg":"fail", "HungMsg":"hung", "EnabledMajorityMsg":true, "HungVoteSetting":4}}

Example 2 :

```
{
  "VotingTemplateAdded": {
    "Id": 3,
    "Title": "Test 223",
    "NrOfOptions": 3,
    "IsPrinted": false,
    "IsVotingNumber": false,
    "BadgeOption": 4,
    "OverallOption": 4,
    "Options": [
      {
        "Id": 66,
        "Description": "",
        "Option": "#FF008000",
        "SymbolIndex": 1,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Accepted",
        "Id": 67,
        "Description": "",
        "Option": "#FFFFFF00",
        "SymbolIndex": 2,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Postponed",
        "Id": 68,
        "Description": "",
        "Option": "#FFFF0000",
        "SymbolIndex": 3,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Rejected"
      }
    ],
    "IndividualOption": 5,
    "CanCorrect": false,
    "HasPin": false,
    "IsWeightUsed": true,
    "IsOperatorIndicated": false,
    "IsSignageIndicated": false,
    "PanelIndicationOption": 0,
    "AgendaTimer": {
      "Id": 109,
      "EnableAgendaTimer": false,
      "DurationOfAgendaTimer": "00:00:00",
      "WarningOfAgendaTimer": "00:00:00",
      "CountDownOfAgendaTimer": true,
      "ExceedOfAgendaTimer": false
    },
    "VotingTimer": {
      "Id": 110,
      "EnableVotingTimer": false,
      "DurationOfVotingTimer": "00:00:00",
      "WarningOfVotingTimer": "00:00:00",
      "CountDownOfVotingTimer": true,
      "ExceedOfVotingTimer": false
    },
    "EnableQuorum": false,
    "AbsoluteOfQuorum": true,
    "ValueOfQuorum": 0.0,
    "OptionOfQuorum": 4,
    "EnableQuorumMsg": false,
    "QuorumMsg": "Quorum not OK - voting postponed",
    "AbsoluteOfOutcome": true,
    "ValueOfMajority": 1.0,
    "MajorityOption": 2,
    "EnableOutcomMsg": false,
    "PassedMsg": "Majority achieved",
    "FailedMsg": "Majority not achieved",
    "HungMsg": "Hung vote",
    "EnabledMajorityMsg": false,
    "HungVoteSetting": 1,
    "IsUnanimous": false
  }
}
```

### 3.2.5 Timer

#### 3.2.5.1 MeetingTimerEvent

API Module	Cocon for Plixus Core	Plixus Core
MeetingTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the current active meeting

**Name:** MeetingTimerEvent

**Parameters:**

Name	Description	Type
MeetingTimer	The timer of current meeting	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{
  "MeetingTimerEvent": {
    "MeetingTimer": {
      "Timer": {
        "TotalTime": "02:00:00",
        "TimeUsed": "00:00:11",
        "CountingDown": true,
        "WarningTime": "00:05:00"
      }
    }
  }
}
```

#### 3.2.5.2 MeetingTimerStateChange

API Module	Cocon for Plixus Core	Plixus Core
MeetingTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or Resume current active meeting Timer

**Name:** MeetingTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
MeetingTimer	The timer of current meeting	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{"MeetingTimerPauseResumed":{"Paused":true,"MeetingTimer":{"Timer":{"TotalTime":"02:00:00","TimeUsed":"00:02:35","CountingDown":true,"WarningTime":"00:05:00"}}}}
```

### 3.2.5.3 DelegateTimerEvent

API Module	Cocon for Plixus Core	Plixus Core
DelegateTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the delegate

**Name:** DelegateTimerEvent

**Parameters:**

Name	Description	Type
DelegateTimer	The time of the delegate	<a href="#">DelegateTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"DelegateTimerEvent":{"DelegateTimer":{"Delegateld":10,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

### 3.2.5.4 DelegateTimerStateChange

API Module	Cocon for Plixus Core	Plixus Core
DelegateTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or resume the delegate timer

**Name:** DelegateTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
DelegateTimer	The timer of the delegate	<a href="#">DelegateTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"DelegateTimerPauseResumed":{"Paused":true,"DelegateTimer":{"Delegateld":10,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

### 3.2.5.5 GroupTimerEvent



API Module	Cocon for Plixus Core	Plixus Core
GroupTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the group

**Name:** GroupTimerEvent

**Parameters:**

Name	Description	Type
GroupTimer	The time of the group	<a href="#">GroupTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"GroupTimerEvent":{"GroupTimer":{"GroupId":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

### 3.2.5.6 GroupTimerStateChange

API Module	Cocon for Plixus Core	Plixus Core
GroupTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or resume the group timer

**Name:** GroupTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
GroupTimer	The timer of the delegate	<a href="#">GroupTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"GroupTimerPauseResumed":{"Paused":true,"GroupTimer":{"GroupId":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

### 3.2.5.7 SeatTimerEvent

API Module	Cocon for Plixus Core	Plixus Core
SeatTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the seat

**Name:** SeatTimerEvent

**Parameters:**

Name	Description	Type
SeatTimer	The timer of the seat	<a href="#">SeatTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"SeatTimerEvent":{"SeatTimer":{"SeatNr":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

**3.2.5.8 SeatTimerPauseResumed**

API Module	Cocon for Plixus Core	Plixus Core
SeatTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or resume the seat timer

**Name:** SeatTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
SeatTimer	The timer of the seat	<a href="#">SeatTimer : BaseObjectTimer</a>

**REST/JON Example:**

```
{"SeatTimerPauseResumed":{"Paused":true,"SeatTimer":{"SeatNr":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}}}}
```

**3.2.5.9 ActiveAgendaItemTimerEvent**

API Module	Cocon for Plixus Core	Plixus Core
ActiveAgendaItemTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the active agenda item

**Name:** ActiveAgendaItemTimerEvent

**Parameters:**

Name	Description	Type
ActiveAgendaItemTimer	The timer of the active agenda item	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{"ActiveAgendaItemTimerEvent":{"ActiveAgendaItemTimer":{"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:21","CountingDown":false,"WarningTime":"00:00:00"}}}}
```

**3.2.5.10 ActiveAgendaItemTimerPauseResumed**

API Module	Cocon for Plixus Core	Plixus Core
ActiveAgendaItemTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or resume the active agenda item timer

**Name:** ActiveAgendaItemTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
ActiveAgendaItemTimer	The timer of the seat	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{"ActiveAgendaItemTimerPauseResumed":{"Paused":false,"ActiveAgendaItemTimer":{"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":false,"WarningTime":"00:00:00"}}}}
```

### 3.2.5.11 VotingTimerEvent

API Module	Cocon for Plixus Core	Plixus Core
VotingTimerEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The timer event of the voting

**Name:** VotingTimerEvent

**Parameters:**

Name	Description	Type
VotingTimer	The timer of the voting	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{"VotingTimerEvent":{"VotingTimer":{"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:02","CountingDown":false,"WarningTime":"00:00:00"}}}}
```

### 3.2.5.12 VotingTimerPauseResumed

API Module	Cocon for Plixus Core	Plixus Core
VotingTimerPauseResumed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pause or resume the voting

**Name:** VotingTimerPauseResumed

**Parameters:**

Name	Description	Type
Paused	The pause state of timer	bool
VotingTimer	The timer of the seat	<a href="#">BaseObjectTimer</a>

**REST/JON Example:**

```
{"VotingTimerPauseResumed":{"Paused":true,"VotingTimer":{"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:32","CountingDown":false,"WarningTime":"00:00:00"}}}}
```

## 3.2.6 Delegate

### 3.2.6.1 BadgeEvent

API Module	Cocon for Plixus Core	Plixus Core
BadgeEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Delegate on the seat, insert badge to show the present. (Fixed seat insert badge for the present meeting mode.)

**Name:** BadgeEvent

**Parameters:**

Name	Description	Type
SeatNr	The number of the seat	int
Delegate	The information of the delegate on seat	<a href="#">Delegate</a>
BadgeInserted	Badge insert or not	bool

**REST/JON Example:**

**Meeting Running :**

```
{"BadgeEvent":{"SeatNr":2,"Delegate":{"Id":463,"FirstName":"Marcel","Name":"Ackermann","MiddleName":null,"Street":"Route Principale 11c","StreetNumber":0,"PostCode":"2803","City":"Bourrignon","Country":"","Title":"Député","BirthDate":"","District":"","Biography":"","Groups":[{"Id":21,"Name":"PDC"}],"SeatNumber":2,"BadgeNumber":"14EFB4E2","FingerprinterData":"","KeypadLoginCode":""},"BadgeInserted":true,"bid":"14EFB4E2"}}
```

**No Meeting running :**

```
{"BadgeEvent":{"SeatNr":4,"Delegate":null,"BadgeInserted":true,"bid":"14EFB4E2"}}
```

Meeting Running :

### 3.2.6.2 DelegateOnSeat

API Module	Cocon for Plixus Core	Plixus Core
DelegateOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>

When seat is empty, and meeting is free-seat, and delegates insert or remove their badges.

**Name:** DelegateOnSeat

**Parameters:**

Name	Description	Type
SeatNr	The number of the seat	int
Delegate	The information of the delegate on seat	<a href="#">Delegate</a>
OnSeat	Delegate is no seat or not	bool

### REST/JON Example:

```
{"DelegateOnSeat":{"SeatNr":4,"Delegate":{"Id":463,"Title":"Député","FirstName":"Marcel","Name":"Ackermann","MiddleName":null,"Groups":[{"Id":21,"Name":"PDC"}]},{"OnSeat":true,"bid":"14EFB4E2"}}
```

### 3.2.6.3 MeetingStartsUserInformation

API Module	Cocon for Plixus Core	Plixus Core
MeetingStartsUserInformation	<input checked="" type="checkbox"/>	<input type="checkbox"/>

When meeting starts, send users in meeting to client.

**Name:** MeetingStartsUserInformation

#### Parameters:

Name	Description	Type
MeetingId	ID of the meeting	int
Users	The Users in meeting	List<Delegate>

### REST/JON Example:

```
{"MeetingStartsUserInformation":{"Users":[{"Id":4,"FirstName":"11","Name":"Test1","Street":"","StreetNumber":null,"PostCode":"","City":"","Country":"","Title":"","BirthDate":"","District":"","Biography":"","Groups":[{"Id":1,"Name":"Group1"}],"SeatNumber":2},{Id":5,"FirstName":"22","Name":"Test2","Street":"","StreetNumber":null,"PostCode":"","City":"","Country":"","Title":"","BirthDate":"","District":"","Biography":"","Groups":[],"SeatNumber":3},{Id":3,"FirstName":"asfwesw","Name":"wefwef","Street":"","StreetNumber":null,"PostCode":"","City":"","Country":"","Title":"Test2","BirthDate":"","District":"","Biography":"","Groups":[{"Id":2,"Name":"Group2"}],"SeatNumber":5},{Id":2,"FirstName":"wefwef","Name":"sdf","Street":"","StreetNumber":null,"PostCode":"","City":"","Country":"","Title":"Test1","BirthDate":"","District":"","Biography":"","Groups":[{"Id":1,"Name":"Group1"}],"SeatNumber":1},{Id":6,"FirstName":"33","Name":"Test3","Street":"","StreetNumber":null,"PostCode":"","City":"","Country":"","Title":"","BirthDate":"","District":"","Biography":"","Groups":[{"Id":2,"Name":"Group2"}],"SeatNumber":4}]}}
```

### 3.2.6.4 DelegateUpdated

API Module	Cocon for Plixus Core	Plixus Core
DelegateUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is sent if a delegate was edited.

**Name:** DelegateUpdated

#### Parameters:

Name	Description	Type
DelegateId	The Id of the delegate	int
Name	The name of the delegate	string
FirstName	The first name of the delegate	string
Title	The title of the delegate	string
MiddleName	MiddleName of the delegate [Optional parameter]	string

BadgeNr	The badge number	int
UserName	The username for the delegate	string
Password	The password for the delegate	string
Street	The street for the delegate	string
StreetNr	The street number for the delegate	string
City	The city for the delegate	string
PostalCode	The postal code for the delegate	string
Country	The country for the delegate	string
District	The district for the delegate	string
BirthDate	The delegate's birthday	string
Email	The delegate's email address	string
PhoneNr	The delegate's phone number	string
VotingRight	Does the delegate have the right to vote?	Bool
VotingWeight	The voting weight to apply for this delegate	Int
PinId	PinId to login for voting	Int
FingerprinterData	Fingerprint data to open the Unicos IDC screen [Optional parameter]	string
KeypadLoginCode	Code to login into the Unicos FMM (IDC) [Optional parameter]	int

#### REST/JON Example:

```
{
  "DelegateUpdated": {
    "DelegateId": 534,
    "Name": "Manoj",
    "MiddleName": "Subhashbhai",
    "FirstName": "Patel",
    "Title": "Mr.",
    "BadgeNr": "",
    "UserName": "",
    "Password": "",
    "Street": "",
    "StreetNr": "0",
    "PostalCode": "",
    "City": "",
    "Country": "",
    "BirthDate": "",
    "District": "",
    "Email": "",
    "PhoneNr": "",
    "VotingRight": true,
    "VotingWeight": 1,
    "PinId": "9876",
    "FingerprinterData": "",
    "KeypadLoginCode": "1234"
  }
}
```

#### 3.2.6.5 DelegateAdded

API Module	Cocon for Plixus Core	Plixus Core
DelegateAdded	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is sent if a delegate was Added.

**Name:** DelegateAdded

#### Parameters:

Name	Description	Type
DelegateId	The Id of the delegate	int
Name	The name of the delegate	string
FirstName	The first name of the delegate	string
Title	The title of the delegate	string
MiddleName	MiddleName of the delegate [Optional parameter]	string
BadgeNr	The badge number	int

UserName	The username for the delegate	string
Password	The password for the delegate	string
Street	The street for the delegate	string
StreetNr	The street number for the delegate	string
City	The city for the delegate	string
PostalCode	The postal code for the delegate	string
Country	The country for the delegate	string
District	The district for the delegate	string
BirthDate	The delegate's birthday	string
Email	The delegate's email address	string
PhoneNr	The delegate's phone number	string
VotingRight	Does the delegate have the right to vote?	Bool
VotingWeight	The voting weight to apply for this delegate	int
PinId	PinId to login for voting	Int
FingerprinterData	Fingerprint data to open the Unicos IDC screen [Optional parameter]	string
KeypadLoginCode	Code to login into the Unicos FMM (IDC) [Optional parameter]	int

#### REST/JON Example:

```
{"DelegateAdded":{"Delegateld":534,"Name":"Isha","MiddleName":null,"FirstName":"Sakhalakr","Title":"Mrs.,"B
adgeNr":"","UserName":"","Password":"","Street":"","StreetNr":0,"PostalCode":"","City":"","Country":"","BirthDat
e":"","District":"","Email":"","PhoneNr":"","VotingRight":true,"VotingWeight":1,"PinId":"","FingerprinterData":"","Ke
ypadLoginCode":""}}
```

### 3.2.6.6 DelegateRemoved

API Module	Cocon for Plixus Core	Plixus Core
DelegateRemoved	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is sent if a delegate was removed from the database.

**Name:** DelegateRemoved

#### Parameters:

Name	Description	Type
Delegateld	The Id of the delegate	int

#### REST/JON Example:

```
{"DelegateRemoved":{"Delegateld":12}}
```

### 3.2.6.7 DelegateSeatingUpdated

API Module	Cocon for Plixus Core	Plixus Core
------------	-----------------------	-------------

DelegateSeatingUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>
------------------------	-------------------------------------	--------------------------

This event is sent to all API clients when the delegate seating is changed (a delegate is added to or removed from a seat).

```
{“DelegateSeatingUpdated”:{“DelegateId”:1, “SeatId”:1}, {“DelegateId”:2, “SeatId”:0}}
```

**Parameters:**

Name	Description	Type
DelegateId	The id of the delegate	int
SeatId	The id of the seat. 0 means that this delegate was not positioned on a seat yet	int

### 3.2.6.8 DelegatesInMeetingUpdated

API Module	Cocon for Plixus Core	Plixus Core
DelegatesInMeetingUpdated	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is sent if something changed to the list of delegates in a meeting (delegate added to or removed from meeting).

Please note that we do not send the new list of delegates in the meeting. API clients that are interested in the change, can request this list with “GetDelegatesInMeeting”.

```
{“DelegatesInMeetingUpdated”:{“MeetingId”:25}}
```

## 3.2.7 Interactive

This section describes commands that are only applicable for the Televic Plixus MME.

### 3.2.7.1 ScreenLockRemoved

API Module	Cocon for Plixus Core	Plixus Core
ScreenLockRemoved	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The screen lock on the uniCOS has been removed

**Name:** ScreenLockRemoved

**REST/JON Example:**

```
{“ScreenLockRemoved” : true}
```

### 3.2.7.2 DelegateScreenSet

API Module	Cocon for Plixus Core	Plixus Core
DelegateScreenSetChanged	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The screen on the Multimedia Unit changed event



**Name:** DelegateScreenSetChanged

**Parameters:**

Name	Description	Type
IsLocked	Lock state of uniCOS	Bool
IsGoTo	Whether changed to the screen	Bool
IsShow	Show or hide the screen	Bool
Screen	Screens on the uniCOS: None Home Discussion Agenda Delegates Voting Documents Messages Web Services Configuration	string
Option	Options for the screen	ScreenControlOptions

**REST/JON Example:**

```
{"DelegateScreenSetChanged" : {"IsLocked": false, "IsGoTo": true, "IsShow": true, "Screen": "Agenda", "Option": {"select": true}}}
```

### 3.2.7.3 CustomButtonEvent

API Module	Cocon for Plixus Core	Plixus Core
CustomButtonEvent	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This event is generated when the custom button is pushed or released.

**REST/JON Example:**

```
{"CustomButtonEvent" : {"SeatId": 5, "ButtonId": 1, "Event": "down"}}
```

## 3.2.8 Audio

### 3.2.8.1 SetAudioSoftwareConfigurationByld

API Module	Cocon for Plixus Core	Plixus Core
SetAudioSoftwareConfigurationByld	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Update event come from the Service & ERS for the SetAudioSoftwareConfigurationByld active preset.

**REST/JON Example:**

```
>>>{"SetAudioSoftwareConfigurationByld":{"AudioSoftwareConfiguration":{"Description":"Incoming audio from microphone, auxiliary inputs 1, 2 & 3 and Dante input channel 1 is routed by default to the unit speakers, auxiliary outputs 1, 2 & 3 and Dante output channel 1.", "Id":1, "Name":"Normal routing", "Predefined":true, "State":"Active"}}
```

### 3.2.8.2 CurrentAudioSoftwareConfigurationUpdated

API Module	Cocon for Plixus Core	Plixus Core
CurrentAudioSoftwareConfigurationUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Update event come from the Server for the volume change for node in active preset.

#### REST/JON Example:

```
"CurrentAudioSoftwareConfigurationUpdated":1}
```

### 3.2.9 Logging

#### 3.2.9.1 ConnectionLost\_CentralUnit

API Module	Cocon for Plixus Core	Plixus Core
Connection_CentralUnit	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Lost central unit connection message

**Name:** Connection\_CentralUnit

#### Parameters:

Name	Description	Type
State	The state of the connection Connected Disconnected	string

#### REST/JON Example:

```
{"Connection_CentralUnit":{"State":<State>}}  
{"Connection_CentralUnit":{"State":"Disconnected"}}
```

#### 3.2.9.2 ConnectionLost\_Database

API Module	Cocon for Plixus Core	Plixus Core
Connection_Database	<input checked="" type="checkbox"/>	<input type="checkbox"/>

CoCon and database connection state

**Name:** Connection\_Database

#### Parameters:

Name	Description	Type
State	The state of the connection Connected Disconnected	string

#### REST/JON Example:

```
{"Connection_Database":{"State":<State>}}
```

### 3.2.10 ButtonLED\_Event

API Module	Cocon for Plixus Core	Plixus Core
ButtonPressed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Events from CoCon Server to API clients can only be sent after the buttons have been enabled, using the command "SetButtonsEnabled", described in section 3.3.11.1.

#### 3.2.10.1 ButtonPressed

API Module	Cocon for Plixus Core	Plixus Core
ButtonPressed	<input checked="" type="checkbox"/>	<input type="checkbox"/>

A button was pressed on a unit.

**Name:** ButtonPressed

#### Parameters:

Name	Description	Type
SeatNr	The number of the seat on which a button was pressed.	int
ButtonNr	The number of the button which was pressed Possible values: 1-5, depending on the possibilities of the delegate units.  For example: delegate units with 3 buttons only have buttons 2-3-4	int

#### REST/JSON Example:

```
{"ButtonPressed":{"SeatNr":2;"ButtonNr":3}}
```

#### 3.2.10.2 NextButtonPressed

API Module	Cocon for Plixus Core	Plixus Core
NextButtonPressed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Next button was pressed on a unit.

**Name:** NextButtonPressed

#### Parameters:

Name	Description	Type
UnitId	The unit's Id	int

#### REST/JSON Example:

```
{"NextButtonPressed":{"UnitId":2}}
```

### 3.2.10.3 PriorButtonPressed

API Module	Cocon for Plixus Core	Plixus Core
PriorButtonPressed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Prior button was pressed on a unit.

**Name:** PriorButtonPressed

**Parameters:**

Name	Description	Type
UnitId	The unit's Id	int

**REST/JSON Example:**

```
{"PriorButtonPressed":{"UnitId":2}}
```

### 3.2.11 Interpretation

#### 3.2.11.1 IMicrophoneStateUpdated

API Module	Cocon for Plixus Core	Plixus Core
IMicrophoneStateUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This function contain the microphone state changed events, channel changed events based on the desk.

Description:

When a microphone event of the ID s happens CoCon sends an update of the realltime data. This contains all the ID microphone statuses. This is not required. Only send the information of the microphone unit that was changed.

**Note:**

Not necessary to subscribe to this module. It will send 3rdpart API based on the interpreter licensed mode. And

**MIC state:** 0 = ON && 2 = OFF.

Ex.

```
{"IMicrophoneStateUpdated":{"Order":2,"Language":"French","Booth":"2","Desk":1,"Micstate":0}}
```

#### 3.2.11.2 IActivatedChannelUpdated

API Module	Cocon for Plixus Core	Plixus Core
IActivatedChannelUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

These events receive by API once the any channel updated on Interpreter desk.

**Note:**

**A/B/C channel index:** 0: Active channel A, 1: active channel B and 2 : active channel C (Interpreter Desk output channel)

Ex: {"IActivatedChannelUpdated":{"Order":2,"Language":"French","Booth":"2","Desk":1,"ActiveChannel":0}}

### 3.2.11.3 SpeakSlowerRequest

API Module	Cocon for Plixus Core	Plixus Core
SpeakSlowerRequest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This event is sent when an interpreter has pressed the slow-down button on his unit.

**Note:** Speak Slow function should be enabled for receiving notification.

#### Parameters:

Name	Description	Type
UnitId	The unit's Id	Int
Booth	Booth id	Int
Desk	Desk id in the booth	int

#### Return:

```
{"SpeakSlowerRequest":{"UnitId":8}}
```

### 3.2.11.4 SpeakSlowerEnded

API Module	Cocon for Plixus Core	Plixus Core
SpeakSlowerEnded	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

These events receive by API once speak slower ended.

#### Return:

```
{"SpeakSlowerEnded":{"UnitId":8}}
```

### 3.2.11.5 ActivePresetStateChanged

API Module	Cocon for Plixus Core	Plixus Core
ActivePresetStateChanged	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.2.11.6

Received event once the active pre-set has been changed.

Note: Bool value received (True : active pre-set has changed.)

Ex: {ActivePresetStateChanged:"true"}

### 3.2.12 Recording ( AE-R & Plixus Core only)

Notification for the recording module.

#### 3.2.12.1 RecordingStatusUpdate

API Module	Cocon for Plixus Core	Plixus Core
WcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Notify the status change of the recording.

**Return: string**

```
{"RecordingStateUpdated":{"RecordingState":"idle"}}
```

### 3.2.13 Wireless Coupling

Notification from the WCAP to rest api user.

#### 3.2.13.1 WcapInfoUpdated

API Module	Cocon for Plixus Core	Plixus Core
WcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Notification for the update from WCAP.

**Return: string**

```
{"WcapInfoUpdated":{"chans":[],"cpld":1,"gtw":"192.168.8.1","hn":"admin","ipa":"192.168.8.210","mac":"1C:BA:8C:99:FE:81","online":1,"op_mode":1,"reg":"Europe","sbm":"255.255.255.0","serial":352321544,"wlist":[]}}
```

#### 3.2.13.2 WcapCoupledUpdated

API Module	Cocon for Plixus Core	Plixus Core
WcapCoupledUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Notification for the update of WCAP coupling from WCAP.

**Return: string**

```
{"WcapCoupledUpdated":{"CoupledMode":true,"WcapSerial":352321544}}
```

#### 3.2.13.3 AllWcapInfoUpdated

API Module	Cocon for Plixus Core	Plixus Core
AllWcapInfoUpdated	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Notification for the update of all WCAP coupling from WCAP.

**Return: string**

```
{"AllWcapInfoUpdated":[{"chans":[],"cpld":1,"gtw":"192.168.8.1","hn":"admin","ipa":"192.168.8.210","mac":"1C:BA:8C:99:FE:81","online":1,"op_mode":0,"reg":"Europe","sbm":"255.255.255.0","serial":352321544,"wlist":[353436539,353436545,353436622]}]}
```

### 3.3 Client to server

For those methods that do not explicitly return a result (usually the methods starting with "Set"), the following return values are used:

- 0: Command was successfully executed.
- 1: parameter or value passed in the API call was wrong. See the CoCon Server log file for more details.
- -1: Exception occurred on the CoCon API side. See the CoCon Server log file for more details.

#### 3.3.1 General

##### 3.3.1.1 Connect

API Module	Cocon for Plixus Core	Plixus Core
Connect	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This is the first URI that has to be sent to initiate the API connection.

**Structure:**

<http://localhost:8890/CoCon/Connect>

**Parameters:** N/A

**Example:**

<http://localhost:8890/CoCon/Connect>

**Return:** {"Connect":<bool>,"id":<id>"}  
{"Connect":true,"id":"970196c9-6f6c-4601-b7d7-f27b1d591bc5"}

##### 3.3.1.2 Subscribe

##### 3.3.1.3 Unsubscribe

API Module	Cocon for Plixus Core	Plixus Core
Subscribe/ UnSubscribe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Subscribe and Unsubscribe are the subscribe/publish mode of our system, by default, once connected to CoCon the client subscribe all modes of cocon, include: Microphone, meetingAgenda, Voting, Timer, Delegate, Audio, Logging.

**Structure:**

<http://localhost:8890/CoCon/Subscribe/?Model={model}&id={id}&details={bool}>

<http://localhost:8890/CoCon/Unsubscribe/?Model={model}&id={id}>

**Parameters:**

Name	Description	Type
Model	The mode that want to subscribe/Unsubscribe Room Microphone	string

	MeetingAgenda Voting Timer Delegate Audio Interpretation Logging ButtonLED_Event Interactive External Intercom Video	
Id	The id that is returned by connection	String
Details	To get all update events information in details *Optional parameter. Default value is true	bool

**Example:**

<http://localhost:8890/CoCon/Subscribe/?Model=Microphone&id=5a44aed8-e257-4b21-9240-efdd72bf866d>  
<http://localhost:8890/CoCon/Unsubscribe/?Model=Microphone&id=a372c5d5-a5ce-4c3a-b401-53d6cd85dbeb>

**Return:** {"Subscribe":{"Model":"Microphone","result":true,"details":true}}  
{"UnSubscribe":{"Model":"Microphone","result":true}}

**3.3.1.4 Notification**

API Module	Cocon for Plixus Core	Plixus Core
Notification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get notification from CoCon, this request should keep asking the CoCon.

**Structure:**

<http://localhost:8890/CoCon/Notification/id={id}>

**Parameters:**

Name	Description	Type
id	The id that is returned by connection	string

**Example:**

<http://localhost:8890/CoCon/Notification/id=5a44aed8-e257-4b21-9240-efdd72bf866d>

**Return:**

without connection: {"Notification" : "Error 400"}  
with connection {"EventName": Event Value}  
with connectin {"Notification" : "Error 408"}

**3.3.1.5 RebootSystem**

API Module	Cocon for Plixus Core	Plixus Core
RebootSystem	<input checked="" type="checkbox"/>	<input type="checkbox"/>



Reboot the CPU.

<http://localhost:8890/CoCon/RebootSystem>

**Return:** int

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.2 Room

#### 3.3.2.1 GetRoomServerName

API Module	Cocon for Plixus Core	Plixus Core
GetRoomServerName	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get the name of the current room server

**Structure:**

<http://localhost:8890/CoCon/Room/GetRoomServerName>

**Example:**

<http://localhost:8890/CoCon/Room/GetRoomServerName>

**Return:**

```
{"GetRoomServerName":{"Name":"Isha Test 250 API"}}
```

#### 3.3.2.2 GetActiveRoomInfo

API Module	Cocon for Plixus Core	Plixus Core
GetActiveRoomInfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get the current room information from the room server.

**Structure:**

<http://192.168.8.126:8890/CoCon/Room/GetActiveRoomInfo>

**Example:**

<http://192.168.8.126:8890/CoCon/Room/GetActiveRoomInfo>

**Return:**

CoCon for Core :

```
{"GetActiveRoomInfo":{"ActiveRoom":{"RoomId":1,"Name":"CoCon 5.3.2.23 (230)","CoConServerVersion":"5.4.0.14","CRPVersion":"5.4.10","DBName":"LOCALHOSTSQLEXPRESS"}}
```

**Note:** RoomId is CoCon DB Room Id.

Plixus Core :

```
{"GetActiveRoomInfo":{"ActiveRoom":{"RoomId":1}}}
```

**Note:** RoomId: active room on Plixus.

### 3.3.2.3 GetStatusOfCuAndDbConnection

API Module	Cocon for Plixus Core	Plixus Core
GetStatusOfCuAndDbConnection	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get the status of the CU connection and DB connection.

**Structure:**

<http://localhost:8890/CoCon/Room/GetStatusOfCuAndDbConnection>

**Example:**

<http://localhost:8890/CoCon/Room/GetStatusOfCuAndDbConnection>

**Return:**

```
{"GetStatusOfCuAndDbConnection":{"CuState":"Connected","DbState":"Connected"}}
```

### 3.3.2.4 SetInitializationState

API Module	Cocon for Plixus Core	Plixus Core
SetInitializationState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the initialization state.

**Structure:**

<http://localhost:8890/CoCon/Room/SetInitializationState/?State={state}>

**Parameters:**

Name	Description	Type
State	The initialization will be set. The state could be like: <b>Reset</b> (Manual init completely) <b>Auto</b> (Auto init completely) <b>Stop</b> <b>Retrieve</b>	string

**Note:**

Only "Reset" and "Stop" working in Cocon for Plixus Core and Plixus Core

**Example:**

<http://localhost:8890/CoCon/Room/SetInitializationState/?State=Auto>

**Return:**

0 : ok

-1: exception at CoCon side

1 : exception at 3rd party side.(please check parameters)

### 3.3.2.5 GetAllUnitsState

API Module	Cocon for Plixus Core	Plixus Core
GetAllUnitsState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all units state which is initialized in CoCon.

**Parameters:**

Name	Description	Type
Roomid	The connected room id on CoCon DB.	int

Return State : 0 = unit is offline 1 = unit is online.

**Structure:**

<http://localhost:8890/CoCon/Room/GetAllUnitsState/?RoomId={roomId}>

**Example:**

<http://localhost:8890/CoCon/Room/GetAllUnitsState/?RoomId=1>

**Return by events:**

```
{"GetAllUnitsState":{"Units":[{"State":1,"UnitId":0},{"State":1,"UnitId":1},{"State":1,"UnitId":2},{"State":1,"UnitId":3}]}}
```

**3.3.2.6 SetDefaultVolumeForRoom**

API Module	Cocon for Plixus Core	Plixus Core
SetDefaultVolumeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set default volume on CU.

**Parameters:**

Name	Description	Type
Room	The connected room on CU (1)	int
Volume	The number of volume (0-25)	int

Return value: int

**Structure:**

<http://localhost:8890/CoCon/Room/SetDefaultVolumeForRoom/?Room={roomNr}&Volume={volume}>

**Example:**

<http://localhost:8890/CoCon/Room/SetDefaultVolumeForRoom/?Room=1&Volume=24>

**Return by events:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

Update event :

```
{"VolumeChangedForRoom":{"Id":1,"Volume":24}}
```

**3.3.2.7 GetDefaultMicrophoneModeForRoom**

API Module	Cocon for Plixus Core	Plixus Core
GetDefaultmicrophoneModeForRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get default mic mode for the active room.

**Structure:**

<http://localhost:8890/CoCon/Room/GetDefaultmicrophoneModeForRoom>

**Example:**

<http://localhost:8890/CoCon/Room/GetDefaultmicrophoneModeForRoom>

**Return by events:**

```
{"GetDefaultMicrophoneModeForRoom":{"MicrophoneMode":{"AllowCancelRequest":true,"AllowSwitchOffMic":true,"FIFO":true,"MaxNrActive":1,"Mode":"DirectSpeak","ReplyRequest":false}}}
```

### 3.3.2.8 SetDefaultMicrophoneModeForRoom

API Module	Cocon for Plixus Core	Plixus Core
SetDefaultmicrophoneModeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the default microphone mode for the room.

**Parameters:**

Name	Description	Type
Mode	The Mode of microphone Operator DirectSpeak Request Vox (Cocon 5.0 :Vox become activation type with Groupdiscussion.) OnlyRequest	string
MaxNrActive	The maximum number of active microphone	int
MaxNrRequest	The maximum number of request microphone	int
AllowRequest	bool indicating whether the delegates are allowed to request. Not applicable for all modes. Optional	bool
AllowCancelRequest	bool indicating whether the delegates are allowed to cancel their request. Not applicable for all modes. Optional	bool
FIFO	bool indicating whether the FIFO mode is active. Not applicable for all modes. Optional	bool
AllowSwitchOffMic	bool indicating whether delegates can switch off their own microphones. Optional	bool

**Structure:**

Basic structure:

<http://localhost:8890/CoCon/Microphone/SetMicrophoneMode/?Mode={mode}&MaxNrActive={MaxNrActive}&MaxNrRequest={MaxNrRequest}>

Optional structure:

<http://localhost:8890/CoCon/Microphone/SetMicrophoneMode/?Mode={mode}&MaxNrActive={maxNrActive}&MaxNrRequest={maxNrReq}&AllowRequest={allowRequest}&AllowCancelRequest={allowCancelRequest}&FIFO={IsFIFO}&AllowSwitchOffMic={allowSwitchOffMic}>

**Example:**

<http://localhost:8890/CoCon/Room/SetDefaultmicrophoneModeForRoom?Mode=DirectSpeak&MaxNrActive=4&MaxNrRequest=200&AllowRequest=False&AllowCancelRequest=False&AllowSwitchOffMic=True&FIFO=true>

**Return by events:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.2.9 SetVolumeForRoom**

API Module	Cocon for Plixus Core	Plixus Core
SetVolumeForRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set volume on CU. This will be temp; once the CU re-start it will back as default value.

**Parameters:**

Name	Description	Type
Room	The connected room on CU (1)	int
Volume	The number of volume (0-25)	int

Return value: int

**Structure:**

<http://localhost:8890/CoCon/Room/SetVolumeForRoom/?Room={roomNr}&Volume={volume}>

**Example:**

<http://localhost:8890/CoCon/Room/SetVolumeForRoom/?Room=1&Volume=24>

**Return by events:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.2.10 AddSynoptic**

API Module	Cocon for Plixus Core	Plixus Core
AddSynoptic	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add new synoptic to current active room.

Note : It will only create new Synoptic if no any synoptic in active room.

**Structure:**

<http://localhost:8890/CoCon/Room/AddSynoptic>

**Example:**

<http://localhost:8890/CoCon/Room/AddSynoptic>

**Return by events:** string

```
{"AddSynoptic":{"RoomSynoptic":{"SynopticId":2,"SynopticName":"Manoj T2_1","IsMaster":true}}
```

### 3.3.2.11 GetAllSeats

API Module	Cocon for Plixus Core	Plixus Core
GetAllSeats	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all seats from the Room.

**Structure:**

<http://localhost:8890/CoCon/Room/GetAllSeats/?RoomId={roomId}>

**Example:**

<http://localhost:8890/CoCon/Room/GetAllSeats/?RoomId=1>

**Return by events:** string

```
{"GetAllSeats":{"Seats":[{"IsChairMan":false,"SeatId":"0","SeatName":"Seat_0","UnitId":"0"}, {"IsChairMan":false,"SeatId":"1","SeatName":"Seat_1","UnitId":"1"}, {"IsChairMan":true,"SeatId":"2","SeatName":"Seat_2","UnitId":"2"}, {"IsChairMan":false,"SeatId":"3","SeatName":"Seat_3","UnitId":"3"}]}}
```

### 3.3.2.12 GetBooths

API Module	Cocon for Plixus Core	Plixus Core
GetBooths	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all boot inform from the current room.

**Structure:**

<http://localhost:8890/CoCon/Room/GetBooths>

**Example:**

<http://localhost:8890/CoCon/Room/GetBooths>

**Return:**

```
{"GetBooths":{"Booths":[{"Order":1,"Units":[{"UnitId":1}]}]}}
```

### 3.3.2.13 EditSeat

API Module	Cocon for Plixus Core	Plixus Core
EditSeat	<input checked="" type="checkbox"/> Note: Not all options are implemented, chairman should work	<input checked="" type="checkbox"/> Note: Not all options are implemented, chairman should work

This api call will allow user to edit seat properties based on the seat number.

Note: After edit seat, user should have to send the SaveRoomConfig command to make sure modification take correct action.

**Parameters:**

Name	Description	Type
SeatNumber	Existing seat number from the active room.	int
Intercom	Is intercom is enabled for the seat (True/ False)	bool
Description	Description	String
IsChairMen	Change seat priority (True/False)	bool

Return value: string

**Structure:**

<http://localhost:8890/CoCon/Room/EditSeat/?SeatNumber={seatNumber}&Intercom={isIntercom}&Description={description}&IsChairMan={IsChairMan}>

**Example:**

<http://localhost:8890/CoCon/Room/EditSeat/?SeatNumber=3&Intercom=True&Description=hi&IsChairMan=True>

**Return by events:** string

```
{"EditSeat":{"SeatId":3,"SeatName":"Seat 3","Description":"hi","UnitId":"","Type":"","IsIntercom":true,"IsChairman":true}}
```

1: for fail / Exception string in log.

### 3.3.2.14 GetAllUnits

API Module	Cocon for Plixus Core	Plixus Core
GetAllUnits	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all Units from the CU.

**Parameters:**

Name	Description	Type
Type	The type of units to retrieve from the CU (e.g. IDesk) Optional	string

Returns: String.

Note:

- return 1 if a) no database or b) no room
- return an empty list in case no units in room (but there is a database and a room)
- return list of units --> normal case

**Structure:**

<http://localhost:8890/CoCon/Room/GetAllUnits>

**Example:**

<http://localhost:8890/CoCon/Room/GetAllUnits>

**Return by events:**

```
{"GetAllUnits":{"Units":[{"RoomId":1,"SystemUnitId":1,"SerialNumber":"15200241","UnitType":"Plixus NE"}, {"RoomId":1,"SystemUnitId":2,"SerialNumber":"153a003d","UnitType":"Confidea T Delegate Unit"}, {"RoomId":1,"SystemUnitId":3,"SerialNumber":"15511567","UnitType":"Confidea F-CS"}, {"RoomId":1,"SystemUnitId":4,"SerialNumber":"141a0007","UnitType":"uniCOS F/MM"}, {"RoomId":1,"SystemUnitId":5,"SerialNumber":"161a0004","UnitType":"uniCOS F/MM 10"}, {"RoomId":1,"SystemUnitId":6,"SerialNumber":"157b034f","UnitType":"Confidea F-DIVC"}, {"RoomId":1,"SystemUnitId":7,"SerialNumber":"157b002d","UnitType":"Confidea F-DIVC"}, {"RoomId":1,"SystemUnitId":8,"SerialNumber":"14200197","UnitType":"Lingua ID"}]}}
```

### 3.3.2.15 SetOperatingMode

API Module	Cocon for Plixus Core	Plixus Core
SetOperatingMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set Operating mode for CU.

Note: On set operating mode need to stop the init by calling “ Room/SetInitializationState/?State=Stop” on the necessary mode.

Ex. If user send this call “Room/SetOperatingMode/?Mode=3&Append=False” they have to send the stop init call to stop the init and it will update the CoCon DB.

#### Parameters:

Name	Description	Type
Mode	Set the Config mode (0 to 4) 0 -> default 1 -> open_access 2 -> from_config (If Append is true and ConfigName is empty, it will just load the current configuration again) 3 -> auto_init 4 -> man_init	int
Append	Set append mode	bool
ConfigName	Set the description of the configuration	string

Returns: int.

#### Structure:

<http://localhost:8890/CoCon/Room/SetOperatingMode/?Mode={mode}&Append={bAppend}&ConfigName={configName}>

#### Example:

<http://localhost:8890/CoCon/Room/SetOperatingMode/?Mode=3&Append=True>

Or

<http://localhost:8890/CoCon/Room/SetOperatingMode/?Mode=3&Append=True&ConfigName=XYZ>

**Return by events:** int

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.2.16 GetOperatingMode

API Module	Cocon for Plixus Core	Plixus Core
GetOperatingMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get Operating mode from CU

#### Structure:

<http://localhost:8890/CoCon/Room/GetOperatingMode>



**Example:**

<http://localhost:8890/CoCon/Room/GetOperatingMode>

**Returns:**

```
{"GetOperatingMode":{"OperatingInfo":{"Mode":2,"Append":true,"ConfigName":"Default Configuration"}}
```

**3.3.2.17 ClearAllMeetingAndDelegateData**

API Module	Cocon for Plixus Core	Plixus Core
ClearAllMeetingAndDelegateData	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This API will clear the all meeting & Meeting template data as well as clear the Delegate list from DB.

Note: this api keep the room configuration as it is. (Ex. Synoptic & units etc.)

**Structure:**

<http://localhost:8890/CoCon/Room/ClearAllMeetingAndDelegateData/?IncludeService={isIncludeService}>

**Parameters:**

Name	Description	Type
IncludeService	Service data can be keep or remove by choice. True = clear service data False = keep service related data. (Optional Parameter)	bool

**Example:**

<http://localhost:8890/CoCon/Room/ClearAllMeetingAndDelegateData>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.3 Microphone****3.3.3.1 SetState**

API Module	Cocon for Plixus Core	Plixus Core
SetState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the state of the microphone

**Structure:**

<http://localhost:8890/CoCon/Microphone/SetState/?State={State}&SeatNr={seatNr}>

**Parameters:**

Name	Description	Type
State	The state of microphone On Off Request Toggle	string
seatNr	The number of the seat	int

**Example:**

<http://localhost:8890/CoCon/Microphone/SetState/?State=On&SeatNr=1>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.3.2 SetMicrophoneMode**

API Module	Cocon for Plixus Core	Plixus Core
SetMicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the mode of the microphone

**Structure:**

Basic structure:

<http://localhost:8890/CoCon/Microphone/SetMicrophoneMode/?Mode={mode}&MaxNrActive={MaxNrActive}&MaxNrRequest={MaxNrRequest}>

Optional structure:

<http://localhost:8890/CoCon/Microphone/SetMicrophoneMode/?Mode={mode}&MaxNrActive={maxNrActive}&MaxNrRequest={maxNrReq}&AllowRequest={allowRequest}&AllowCancelRequest={allowCancelRequest}&FIFO={IsFIFO}&AllowSwitchOffMic={allowSwitchOffMic}>

**Parameters:**

Name	Description	Type
Mode	The Mode of microphone Operator DirectSpeak Request Vox (Cocon 5.0 :Vox become activation type with Groupdiscussion.) OnlyRequest	string
MaxNrActive	The maximum number of active microphone	int
MaxNrRequest	The maximum number of request microphone	int
AllowRequest	bool indicating whether the delegates are allowed to request. Not applicable for all modes. Optional	bool
AllowCancelRequest	bool indicating whether the delegates are allowed to cancel their request. Not applicable for all modes. Optional	bool
FIFO	bool indicating whether the FIFO mode is active Not applicable for all modes. Optional	bool
AllowSwitchOffMic	bool indicating whether delegates can switch off their own microphones. Optional	bool

**Example:**

<http://localhost:8890/CoCon/Microphone/SetMicrophoneMode/?Mode=OnlyRequest&MaxNrActive=4&MaxNrRequest=200&AllowRequest=False&AllowCancelRequest=False&AllowSwitchOffMic=True>

**Return by events:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.3.3 SetActivationType**

API Module	Cocon for Plixus Core	Plixus Core
SetActivationType	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set activation type is used for the change the microphone activation type available on the current microphone mode.

**Structure:**

<http://localhost:8890/CoCon/Microphone/SetActivationType/?ActivationType={activationType}>

**Parameters:**

Name	Description	Type
ActivationType	Optional User can set the activation type base on the active mic mode. Ex: Direct speaker : "Push Or Toggle" Request(Group discussion) : "Push Or Toggle Or Vox".	string

**Example:**

<http://192.168.8.250:8890/CoCon/Microphone/SetActivationType/?ActivationType=Toggle>

**Return by events:**

**Int:** 0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.3.4 SetMicrophoneSensitivity**

API Module	Cocon for Plixus Core	Plixus Core
SetMicrophoneSensitivity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set microphone sensitivity for the seat (Microphone).

**Structure:**

<http://localhost:8890/CoCon/Microphone/SetMicrophoneSensitivity/?SeatId={seatId}&Sensitivity={sensitivity}>

**Parameters:**

Name	Description	Type
SeatId	The number of the seat	int
Sensitivity	Value of the sensitivity Range of value +12 to -12 Ex. (12,11,10,9...-10,-11,-12)	int

**REST/JON Example:**

<http://localhost:8890/CoCon/Microphone/SetMicrophoneSensitivity/?SeatId=1&Sensitivity=0>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.3.5 Get

API Module	Cocon for Plixus Core	Plixus Core
Get	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all information of Cocon

**Structure:**

<http://localhost:8890/CoCon/Microphone/Get>

**Example:**

<http://localhost:8890/CoCon/Microphone/Get>

**Return:**

```
{ "Get" : { "MicrophoneMode" : { "Mode" : "Operator", "MaxNrActive" : 4, "MaxNrRequest" : 1, "State" : { "Speakers" : [], "Requests" : [], "Replies": [] } } }
```

**Note:**MaxNrRequest is no longer a part of the reply parameter for CoCon for Plixus Core & PlixusCore.

```
{ "Get":{ "MicrophoneMode":{ "MaxNrActive":4, "Mode":"Request"}, "State":{ "Requests":[], "Speakers":[] } }
```

### 3.3.3.6 GetState

API Module	Cocon for Plixus Core	Plixus Core
GetState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all microphones state of CoCon

**Structure:**

<http://localhost:8890/CoCon/Microphone/GetState>

**Example:**

<http://localhost:8890/CoCon/Microphone/GetState>

**Return:**

```
{ "GetState":{ "State":{ "Speakers":[5], "Requests":[4], "Replies":[] } } }
```

### 3.3.3.7 ClearMicrophoneList

API Module	Cocon for Plixus Core	Plixus Core
ClearMicrophoneList	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Clear microphone list.

**Structure:**

<http://localhost:8890/CoCon/Microphone/ClearMicrophoneList/?Type={type}>

**Parameters:**

Name	Description	Type
Type	SpeakersAndRequests (Include Chiarman) Speakers (not Include chairman) Requests	string

**Example:**

<http://localhost:8890/CoCon/Microphone/ClearMicrophoneList/?Type=SpeakersAndRequests>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.3.8 GetMicrophoneMode

API Module	Cocon for Plixus Core	Plixus Core
GetMicrophoneMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get microphone mode of CoCon

**Structure:**

<http://localhost:8890/CoCon/Microphone/GetMicrophoneMode>

**Example:**

<http://localhost:8890/CoCon/Microphone/GetMicrophoneMode>

**Return:**

```
{"GetMicrophoneMode":{"MicrophoneMode":{"AllowCancelRequest":true,"AllowSwitchOffMic":true,"FIFO":true,"MaxNrActive":4,"Mode":"Request","ReplyRequest":false}}}
```

### 3.3.3.9 ActivateNextMicrophone

API Module	Cocon for Plixus Core	Plixus Core
ActivateNextMicrophone	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Activate the next microphone in the request list

**Structure:**

<http://localhost:8890/CoCon/Microphone/ActivateNextMicrophone>

**Example:**

<http://localhost:8890/CoCon/Microphone/ActivateNextMicrophone>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

## 3.3.4 Meeting\_Agenda

### 3.3.4.1 SetMeetingState

API Module	Cocon for Plixus Core	Plixus Core
SetMeetingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set the state of selected meeting

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetMeetingState/?State=<State>&MeetingId=<MeetingId>](http://localhost:8890/CoCon/Meeting_Agenda/SetMeetingState/?State=<State>&MeetingId=<MeetingId>)

**Parameters:**

Name	Description	Type
State	The state of meeting Running Ended Paused	string
MeetingId	The id of the meeting	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetMeetingState/?State=Running&MeetingId=8](http://localhost:8890/CoCon/Meeting_Agenda/SetMeetingState/?State=Running&MeetingId=8)

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.4.2 StartEmptyMeeting**

API Module	Cocon for Plixus Core	Plixus Core
StartEmptyMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Start an empty meeting directly

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/StartEmptyMeeting/?Title=<meetingTitle>&From=<meetingFrom>&To=<meetingTo>&LoginMethod=<loginMethod>](http://localhost:8890/CoCon/Meeting_Agenda/StartEmptyMeeting/?Title=<meetingTitle>&From=<meetingFrom>&To=<meetingTo>&LoginMethod=<loginMethod>)

**Parameters:**

Name	Description	Type
Title	The title of the empty meeting	string
From	The time of the meeting to be started (the time must be a future time)	DateTime
To	The time of the meeting to be finished (the time must be after From time)	DateTime
LoginMethod	(Option field) Default login method = 2 0 = Fixed seat + insert Badge 1 = Fixed seat ignore badge. 2 = Free seat meeting.	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/StartEmptyMeeting/?Title=This is a empty meeting&From=2017/6/5 00:00:00&To=2017/6/5 00:10:00&LoginMethod=1](http://localhost:8890/CoCon/Meeting_Agenda/StartEmptyMeeting/?Title=This is a empty meeting&From=2017/6/5 00:00:00&To=2017/6/5 00:10:00&LoginMethod=1)

**Return:**

[{"StartEmptyMeeting":{"Result":true,"MeetingId":3,"MeetingTitle":"This is a empty meeting","NewMeetingFrom":"2017/06/05 00:00:00","NewMeetingTo":"2017/06/05 00:10:00","LoginMethod":1}}](#)

**3.3.4.3 GetMeetingsForToday**

API Module	Cocon for Plixus Core	Plixus Core
GetMeetingsForToday	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Return all meeting scheduled for today

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetMeetingsForToday](http://localhost:8890/CoCon/Meeting_Agenda/GetMeetingsForToday)

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetMeetingsForToday](http://localhost:8890/CoCon/Meeting_Agenda/GetMeetingsForToday)

**Return:**

```
{"GetMeetings":{"Meetings":[{"Id":1,"Title":"Fixed seat metign ","Description":"Fixed seat metign ","StartTime":"2018/12/10 10:18:32","State":"Ended","Nameplate_Layout":null}, {"Id":3,"Title":"Title hjhjhj","Description":"","StartTime":"2018/12/10 13:49:47","State":"Ended","Nameplate_Layout":null}, {"Id":4,"Title":"This is a empty meeting","Description":"","StartTime":"2018/12/10 13:52:32","State":"Running","Nameplate_Layout":""}]}}
```

### 3.3.4.4 GetAllMeetingsForDate

API Module	Cocon for Plixus Core	Plixus Core
GetAllMeetingsForDate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Return all the meetings for a period of 24 hours after the given Date/Time combination

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAllMeetingsForDate/?Time=<Time>](http://localhost:8890/CoCon/Meeting_Agenda/GetAllMeetingsForDate/?Time=<Time>)

Time format : YYYY/MM/DD HH:mm:ss

Note that the HH:mm:ss can be omitted to retrieve all the meetings taking place on the given date.

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAllMeetingsForDate/?Time=2018/06/08](http://localhost:8890/CoCon/Meeting_Agenda/GetAllMeetingsForDate/?Time=2018/06/08)

**Return:**

```
{"GetMeetings":{"Meetings":[{"Id":2,"Title":"This is a empty meeting","Description":"","StartTime":"2018/12/10 13:52:17","State":"Running","Nameplate_Layout":""}]}}
```

**Parameters:**

Name	Description	Type
Time	The date time (YYYY/MM/DD HH:mm:ss)	DateTime

### 3.3.4.5 GetMeetingById

API Module	Cocon for Plixus Core	Plixus Core
GetMeetingById	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Returns the meeting identified by the certain MeetingId.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetMeetingById/?MeetingId=<MeetingId>](http://localhost:8890/CoCon/Meeting_Agenda/GetMeetingById/?MeetingId=<MeetingId>)

**Parameters:**

Name	Description	Type
MeetingId	The id of the meeting to get	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetMeetingById/?MeetingId=497](http://localhost:8890/CoCon/Meeting_Agenda/GetMeetingById/?MeetingId=497)

**Return:**

```
{"GetMeeting":{"Meeting":{"Id":497,"Title":"Title dtrte","Description":"","StartTime":"2017/08/29 12:53:47","State":"Ended"}}
```

```
{"GetMeeting":{"Meeting":{"Id":2,"Title":"This is a empty meeting","Description":"","StartTime":"2018/12/10 13:52:17","State":"Running","Nameplate_Layout":""}}}
```

**3.3.4.6 SetActiveAgendaItemByDirection**

API Module	Cocon for Plixus Core	Plixus Core
SetActiveAgendaItemByDirection	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Active agenda item by next and previous command

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetActiveAgendaItemByDirection/?Direction=<Direction>](http://localhost:8890/CoCon/Meeting_Agenda/SetActiveAgendaItemByDirection/?Direction=<Direction>)

**Parameters:**

Name	Description	Type
Direction	The direction of active agenda item Next Previous	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetActiveAgendaItemByDirection/?Direction=Next](http://localhost:8890/CoCon/Meeting_Agenda/SetActiveAgendaItemByDirection/?Direction=Next)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.4.7 SetActiveAgendaItemById**

API Module	Cocon for Plixus Core	Plixus Core
SetActiveAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Active agenda item by Id of the agenda item

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetActiveAgendaItemById/?Id=<Id>](http://localhost:8890/CoCon/Meeting_Agenda/SetActiveAgendaItemById/?Id=<Id>)

**Parameters:**

Name	Description	Type
Id	The Id of the agenda item to activate	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/SetActiveAgendaItemById/?Id=3](http://localhost:8890/CoCon/Meeting_Agenda/SetActiveAgendaItemById/?Id=3)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.4.8 GetAgendaItemInformationInRunningMeeting**



API Module	Cocon for Plixus Core	Plixus Core
GetAgendaltemInformationInRunningMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all the agenda item information in running meeting

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaltemInformationInRunningMeeting](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaltemInformationInRunningMeeting)

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaltemInformationInRunningMeeting](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaltemInformationInRunningMeeting)

**Return:** string

Reply with the all agenda in running meeting.

```

{"GetAgendaltemInformationInRunningMeeting":{"Agendaltems":[{"Id":"1","Title":"Test Discussion
1","Description":"Test Discussion
1","Type":"Agendaltem","State":"notstarted","IdInDb":16,"Children":[]},{"Id":"2","Title":"Test Voting
1_3_Vote_Public","Description":"Test Voting
1_3_Vote_Public","Type":"VotingAgendaltem","State":"active","VotingOptions":[{"Id":45,"Name":"","Color":"#FF0
08000"},{"Id":46,"Name":"","Color":"#FFFFFF00"},{"Id":47,"Name":"","Color":"#FFFF0000"}],"Children":[{"Id":"2.1
","Title":"Test Voting 2_2_Vote_Public","Description":"Test Voting
2_2_Vote_Public","Type":"VotingAgendaltem","State":"notstarted","VotingOptions":[{"Id":40,"Name":"","Color":"#
FF008000"},{"Id":41,"Name":"","Color":"#FFFF0000"}],"Children":[],"IdInDb":18},"IdInDb":17}}]}

```

### 3.3.4.9 GetAgendaltemInformationInMeetingById

API Module	Cocon for Plixus Core	Plixus Core
GetAgendaltemInformationInMeetingById	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all the agenda item information in the meeting identified by a given MeetingId.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaltemInformationInMeetingById/?MeetingId=<MeetingId>](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaltemInformationInMeetingById/?MeetingId=<MeetingId>)

**Parameters:**

Name	Description	Type
MeetingId	The Id of the meeting	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaltemInformationInMeetingById/?MeetingId=498](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaltemInformationInMeetingById/?MeetingId=498)

**Return:** string

Reply with the all agenda in running meeting.

```

{"GetAgendaltemInformationInMeetingById":{"Agendaltems":[{"Id":"1","Title":"Test Discussion
1","Description":"Test Discussion
1","Type":"Agendaltem","State":"notstarted","IdInDb":16,"Children":[]},{"Id":"2","Title":"Test Voting
1_3_Vote_Public","Description":"Test Voting
1_3_Vote_Public","Type":"VotingAgendaltem","State":"ended","VotingOptions":[{"Id":45,"Name":"","Color":"#FF0
08000"},{"Id":46,"Name":"","Color":"#FFFFFF00"},{"Id":47,"Name":"","Color":"#FFFF0000"}],"Children":[{"Id":"2.1
","Title":"Test Voting 2_2_Vote_Public","Description":"Test Voting
2_2_Vote_Public","Type":"VotingAgendaltem","State":"active","VotingOptions":[{"Id":40,"Name":"","Color":"#FF0
08000"},{"Id":41,"Name":"","Color":"#FFFF0000"}],"Children":[],"IdInDb":18},"IdInDb":17}}]}

```

### 3.3.4.10 GetAgendaItemById

API Module	Cocon for Plixus Core	Plixus Core
GetAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get the agenda item by agenda item id.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaItemById/?Id=<Id>](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaItemById/?Id=<Id>)

**Parameters:**

Name	Description	Type
Id	The Id of the agenda item	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetAgendaItemById/?Id=2](http://localhost:8890/CoCon/Meeting_Agenda/GetAgendaItemById/?Id=2)

**Return:** string

```
{"GetAgendaItemById":{"AgendaItem":{"Id":"2","Title":"Test Voting 1_3_Vote_Public","Description":"Test Voting 1_3_Vote_Public","Type":"VotingAgendaItem","State":"ended","VotingOptions":[{"Id":45,"Name":"","Color":"#FF008000"},{"Id":46,"Name":"","Color":"#FFFFFF00"},{"Id":47,"Name":"","Color":"#FFF000"}],"Children":[{"Id":"2.1","Title":"Test Voting 2_2_Vote_Public","Description":"Test Voting 2_2_Vote_Public","Type":"VotingAgendaItem","State":"active","VotingOptions":[],"Children":[],"IdInDb":18},"IdInDb":17}}}
```

### 3.3.4.11 StartMeetingViaMeetingTemplate

API Module	Cocon for Plixus Core	Plixus Core
StartMeetingViaMeetingTemplate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Start the meeting via template meeting name

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/StartMeetingViaMeetingTemplate/?TemplateName={templateName}](http://localhost:8890/CoCon/Meeting_Agenda/StartMeetingViaMeetingTemplate/?TemplateName={templateName})

**Parameters:**

Name	Description	Type
TemplateName	Name of the existing meeting template	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/StartMeetingViaMeetingTemplate/?TemplateName=Test meeting](http://localhost:8890/CoCon/Meeting_Agenda/StartMeetingViaMeetingTemplate/?TemplateName=Test meeting)

**Return:**

```
{"StartMeetingViaMeetingTemplate":{"Result":true,"MeetingId":500}}
new created meetingId
```

### 3.3.4.12 StartMeetingViaXMLImport

API Module	Cocon for Plixus Core	Plixus Core
------------	-----------------------	-------------

StartMeetingViaXMLimport	<input checked="" type="checkbox"/>	<input type="checkbox"/>
--------------------------	-------------------------------------	--------------------------

Start the meeting from an imported XML template. The XML file, containing the meeting data to be imported, should be uploaded to an integrated FTP-server, hosted by the CoCon Server.

FTP Server IP: CoCon Server IP  
 Port: 2125  
 User Name: televic  
 Password: tl  
 Path: default path.

Hint: if you want to put the file directly on to the CoCon Server, the actual path on the CoCon Server PC (for Windows 7) is:

C:\Users\<CoCon Server login user>\AppData\Local\Televic Conference\CoCon\Doc\Server\

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/StartMeetingViaXMLimport/?XMLName={xml\\_name}&Type={type}](http://localhost:8890/CoCon/Meeting_Agenda/StartMeetingViaXMLimport/?XMLName={xml_name}&Type={type})

**Return:**

new created meetingId, or -1 when the import has failed

**Parameters:**

Name	Description	Type
XMLName	The name of the XML file to be used for import	string
Type	<p>The type of the XML file for import. Possible values:</p> <p>“Generated”: the XML-format used by the Save Offline function in the Meeting Manager  <b>Note</b> : Group list or Grouprelated info should not be a part of generated xml meeting else meeting start will fail.</p> <p>“Readable”: the XML format as described in section 10.6.3 “Readable XML” of the CoCon manual</p>	string

**Example:**

[http://192.168.8.171:8890/CoCon/Meeting\\_Agenda/StartMeetingViaXMLimport/?XMLName=test-meeting.xml&Type=Readable](http://192.168.8.171:8890/CoCon/Meeting_Agenda/StartMeetingViaXMLimport/?XMLName=test-meeting.xml&Type=Readable)

**3.3.4.13 AddAgendaItem**

API Module	Cocon for Plixus Core	Plixus Core
AddAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add a new agenda item to the agenda.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddAgendaItem/?Title={title}&Des={description}&Sequence={sequence}&Type={type}&VotingTemplate={template}](http://localhost:8890/CoCon/Meeting_Agenda/AddAgendaItem/?Title={title}&Des={description}&Sequence={sequence}&Type={type}&VotingTemplate={template})

**Parameters:**

Name	Description	Type
Title	The title of agenda item	string
Des	The description of agenda item	string
Sequence	The sequence string of agenda item, eg: 1.2.1	string
Type	The type of agenda item: Discussion Lecture Voting	string
VotingTemplate (optional)	The template of voting (optional, if Type is Voting)	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddAgendaltem/?Title=Test Agedna 1&Des=Test Agedna 1&Sequence=2.1&Type=Discussion](http://localhost:8890/CoCon/Meeting_Agenda/AddAgendaltem/?Title=Test Agedna 1&Des=Test Agedna 1&Sequence=2.1&Type=Discussion)

**Return:**

```
{"AddNewAgendaltem":{"Id":"2.1"}}
```

### 3.3.4.14 GetLoginMethodForMeeting

API Module	Cocon for Plixus Core	Plixus Core
GetLoginMethodForMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get login method for the delegates in a meeting

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetLoginMethodForMeeting/?MeetingId=<MeetingId>](http://localhost:8890/CoCon/Meeting_Agenda/GetLoginMethodForMeeting/?MeetingId=<MeetingId>)

**Parameters:**

Name	Description	Type
MeetingId	The Id of the meeting	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetLoginMethodForMeeting/?MeetingId=500](http://localhost:8890/CoCon/Meeting_Agenda/GetLoginMethodForMeeting/?MeetingId=500)

**Return:** string

A string which is one of the following Strings:

1. FixedSeatWithBadge
2. FixedSeatWithoutBadge
3. FreeSeating
4. ""(Empty string,It means get login method failure)

```
{"GetLoginMethodForMeeting":"FreeSeating"}
{"GetLoginMethodForMeeting":"FixedSeatWithoutBadge"}
```

### 3.3.4.15 RemoveAgendaltemById

API Module	Cocon for Plixus Core	Plixus Core
------------	-----------------------	-------------

RemoveAgendaItemById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
----------------------	-------------------------------------	--------------------------

Remove the agenda item by agenda item id.  
Note: It will delete the child item as well.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveAgendaItemById/?Id=<Id>](http://localhost:8890/CoCon/Meeting_Agenda/RemoveAgendaItemById/?Id=<Id>)

**Parameters:**

Name	Description	Type
Id	The Id of the agenda item	string

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveAgendaItemById/?Id=2](http://localhost:8890/CoCon/Meeting_Agenda/RemoveAgendaItemById/?Id=2)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.4.16 DeleteAllAgendaItems

API Module	Cocon for Plixus Core	Plixus Core
DeleteAllAgendaItems	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Delete all the agenda items in meeting

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/DeleteAllAgendaItems/?Id=<meetingId>](http://localhost:8890/CoCon/Meeting_Agenda/DeleteAllAgendaItems/?Id=<meetingId>)

**Parameters:**

Name	Description	Type
Id	The Id of the meeting	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/DeleteAllAgendaItems/?Id=500](http://localhost:8890/CoCon/Meeting_Agenda/DeleteAllAgendaItems/?Id=500)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.4.17 MoveAgendaItem

API Module	Cocon for Plixus Core	Plixus Core
MoveAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Move agenda item location; it provides four type of movement, Up/Down/LevelUp/LevelDown

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/MoveAgendaItem/?Id=<agendaId>&Move=<movement>](http://localhost:8890/CoCon/Meeting_Agenda/MoveAgendaItem/?Id=<agendaId>&Move=<movement>)

**Parameters:**

Name	Description	Type
Id	The Id of the agenda item	string

Move	Include four options (Up/Down/LevelUp/LevelDown)	string
------	---	--------

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/MoveAgendaItem/?Id=2.1&Move=LevelUp](http://localhost:8890/CoCon/Meeting_Agenda/MoveAgendaItem/?Id=2.1&Move=LevelUp)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.4.18 EditAgendaItem

API Module	Cocon for Plixus Core	Plixus Core
EditAgendaItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Edit agenda item based on the order for the running meeting.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/EditAgendaItem/?Id=<id>&Title={title}&Des={Des}](http://localhost:8890/CoCon/Meeting_Agenda/EditAgendaItem/?Id=<id>&Title={title}&Des={Des})

**Parameters:**

Name	Description	Type
Id	The Id(order / sequence id) of the agenda.	string
Title	Title user defined	string
Des	Description for the agenda.	string

Return : String.

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/EditAgendaItem/?Id=1.1&Title=test agenda item&Des=New agenda item](http://localhost:8890/CoCon/Meeting_Agenda/EditAgendaItem/?Id=1.1&Title=test agenda item&Des=New agenda item)

Reply : agenda item with data.

```
{"EditedAgendaItem":{"Id":"1.1","Title":"TestTitle","Des":"Description of the item."}}
```

### 3.3.4.19 AddDelegateToMeeting

API Module	Cocon for Plixus Core	Plixus Core
AddDelegateToMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add delegate to the meeting.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegateToMeeting/?DelegateId={DelegateId}&MeetingId={MeetingId}](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegateToMeeting/?DelegateId={DelegateId}&MeetingId={MeetingId})

**Parameters:**

Name	Description	Type
DelegateId	The Delegate id from DB	int

MeetingId	Meeting id.	int
-----------	-------------	-----

Return : int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegateToMeeting/?DelegatId=43&MeetingId=11](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegateToMeeting/?DelegatId=43&MeetingId=11)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

{"DelegatesInMeetingUpdated":{"MeetingId":11}}

**3.3.4.20 AddDelegatesToMeeting**

API Module	Cocon for Plixus Core	Plixus Core
AddDelegateToMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add delegates to the meeting. (Multiple delegates can be added at same time)

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegatesToMeeting/?DelegatesId={DelegatesId}&MeetingId={MeetingId}](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegatesToMeeting/?DelegatesId={DelegatesId}&MeetingId={MeetingId})

**Parameters:**

Name	Description	Type
DelegatesId	The Delegates id from DB	string
MeetingId	Meeting id.	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegatesToMeeting/?DelegatesId=487,486,485&MeetingId=483](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegatesToMeeting/?DelegatesId=487,486,485&MeetingId=483)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

0, success but these delegates: 487 486 failed, because they not exist in the database

{"DelegatesInMeetingUpdated":{"MeetingId":11}}

**3.3.4.21 RemoveDelegateFromMeeting**

API Module	Cocon for Plixus Core	Plixus Core
RemoveDelegateFromMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Remove delegate from the meeting.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveDelegateFromMeeting/?DelegatId={DelegatId}&MeetingId={MeetingId}](http://localhost:8890/CoCon/Meeting_Agenda/RemoveDelegateFromMeeting/?DelegatId={DelegatId}&MeetingId={MeetingId})

**Parameters:**

Name	Description	Type
------	-------------	------

Delegateld	The Delegate id from DB	int
MeetingId	Meeting id.	int

Return : int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveDelegateFromMeeting/?Delegateld=43&MeetingId=11](http://localhost:8890/CoCon/Meeting_Agenda/RemoveDelegateFromMeeting/?Delegateld=43&MeetingId=11)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

{"DelegatesInMeetingUpdated":{"MeetingId":11}}

**3.3.4.22 GetDelegateSeating**

API Module	Cocon for Plixus Core	Plixus Core
GetDelegateSeating	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get delegate seating form the synoptic.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetDelegateSeating/?MeetingId={MeetingId}](http://localhost:8890/CoCon/Meeting_Agenda/GetDelegateSeating/?MeetingId={MeetingId})

**Parameters:**

Name	Description	Type
MeetingId	Meeting id.	int

Return : int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/GetDelegateSeating/?MeetingId=11](http://localhost:8890/CoCon/Meeting_Agenda/GetDelegateSeating/?MeetingId=11)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

{"GetDelegateSeating":{"DelegateSeating":[{"Delegateld":44,"SeatId":0},{"Delegateld":43,"SeatId":1},{"Delegateld":45,"SeatId":0}]}}

**3.3.4.23 AddDelegateToSeat**

API Module	Cocon for Plixus Core	Plixus Core
AddDelegateToSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add Delegate to seat.

Note: if delegate is in DB that will be added to meeting user(if not a meeting user) and set on seat.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegateToSeat/?MeetingId=12&Delegateld=45&SeatId=4](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegateToSeat/?MeetingId=12&Delegateld=45&SeatId=4)

**Parameters:**

Name	Description	Type
------	-------------	------



MeetingId	Meeting id	int
Delegateld	Delegate id from db	int
SeatId	Seat id on room (Synoptic)	int

Return : int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegateToSeat/?MeetingId=13&Delegateld=45&SeatId=2](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegateToSeat/?MeetingId=13&Delegateld=45&SeatId=2)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

{"DelegatesInMeetingUpdated":{"MeetingId":13}}

**3.3.4.24 AddDelegatesToSeats**

API Module	Cocon for Plixus Core	Plixus Core
AddDelegatesToSeats	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add Delegates to seats API call will allow to assign delegates on seats like list as one time.  
Note: if delegate is in DB that will be added to meeting user (if not a meeting user) and set on seat.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegatesToSeats/?MeetingId={meetingId}&Delegatelds=1,2,3&SeatIds=2,3,4](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegatesToSeats/?MeetingId={meetingId}&Delegatelds=1,2,3&SeatIds=2,3,4)

**Parameters:**

Name	Description	Type
MeetingId	Meeting id	int
Delegatelds	Delegate id from db	int
SeatIds	Seat id on room (Synoptic)	int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/AddDelegatesToSeats/?MeetingId=13&Delegatelds=2,101,125,3&SeatIds=2,3,4](http://localhost:8890/CoCon/Meeting_Agenda/AddDelegatesToSeats/?MeetingId=13&Delegatelds=2,101,125,3&SeatIds=2,3,4)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

{"DelegatesInMeetingUpdated":{"MeetingId":13}}

**3.3.4.25 RemoveDelegateFromSeat**

API Module	Cocon for Plixus Core	Plixus Core
RemoveDelegateFromSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Remove the delegate from seat.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveDelegateFromSeat/?MeetingId={MeetingId}&Delegateld={delegateld}](http://localhost:8890/CoCon/Meeting_Agenda/RemoveDelegateFromSeat/?MeetingId={MeetingId}&Delegateld={delegateld})

**Parameters:**

Name	Description	Type
MeetingId	Meeting id.	int
Delegateld	The Delegate id from DB	int

Return : int

**Example:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/RemoveDelegateFromSeat/?MeetingId=12&Delegateld=43](http://localhost:8890/CoCon/Meeting_Agenda/RemoveDelegateFromSeat/?MeetingId=12&Delegateld=43)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.4.26 ApplyDelegatesToSystem**

API Module	Cocon for Plixus Core	Plixus Core
ApplyDelegatesToSystem	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Apply the seat modification to H/W.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/ApplyDelegatesToSystem](http://localhost:8890/CoCon/Meeting_Agenda/ApplyDelegatesToSystem)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.4.27 EndActiveMeeting**

API Module	Cocon for Plixus Core	Plixus Core
EndActiveMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

End the current active meeting.

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/EndActiveMeeting](http://localhost:8890/CoCon/Meeting_Agenda/EndActiveMeeting)

**Returns:**

{"EndActiveMeeting":{"MeetingId":12}}

**3.3.4.28 ResendMeetingData**

API Module	Cocon for Plixus Core	Plixus Core
ResendMeetingData	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Resend meeting data for current active meeting. (helping to open the IDC log in screen if it not open after restart IDC and delegate already on seat).

**Structure:**

[http://localhost:8890/CoCon/Meeting\\_Agenda/ResendMeetingData](http://localhost:8890/CoCon/Meeting_Agenda/ResendMeetingData)

**Returns:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.5 Voting

#### 3.3.5.1 SetVotingState

API Module	Cocon for Plixus Core	Plixus Core
SetVotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set the voting state of current active voting agenda item

**Structure:**

<http://localhost:8890/CoCon/Voting/SetVotingState/?State=<State>>

**Parameters:**

Name	Description	Type
State	The state of voting Start Stop Pause Resume Restart Clear	string

**Following voting states and transitions are allowed:**

Start : VotingIdle {means voting is clear}

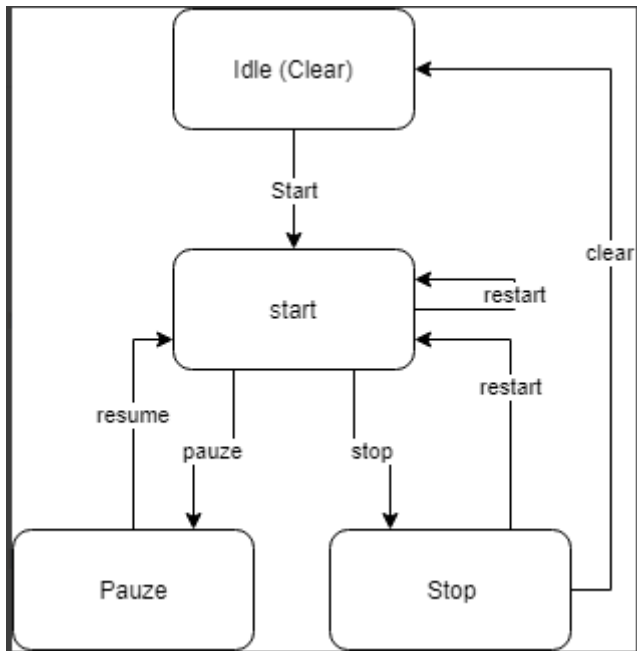
Stop : VotingPaused and VotingStarted

Pause : VotingStarted

Resume : VotingPaused

Clear : VotingStopped

Restart : VotingStarted and VotingStopped



**Example:**

<http://localhost:8890/CoCon/Voting/SetVotingState/?State=Start>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.5.2 AddInstantVote**

API Module	Cocon for Plixus Core	Plixus Core
AddInstantVote	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Adds a new instance voting item

**Structure:**

<http://localhost:8890/CoCon/Voting/AddInstantVote/?VotingTemplate=<VotingTemplate>>

**Parameters:**

Name	Description	Type
VotingTemplate	The voting template name that instance voting based on	string

**Example:**

[http://localhost:8890/CoCon/Voting/AddInstantVote/?VotingTemplate=3\\_Vote\\_Public](http://localhost:8890/CoCon/Voting/AddInstantVote/?VotingTemplate=3_Vote_Public)

**Return:**

{"AddInstantVote":{"Result":true}}

**3.3.5.3 AddVotingTemplate**

API Module	Cocon for Plixus Core	Plixus Core

AddVotingTemplate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
-------------------	-------------------------------------	--------------------------

Will allow user to add new voting template.

Note: Some special character need to replace with other

Character Escape Code		Character Escape Code	
SPACE	%20	[	%5B
<	%3C	]	%5D
>	%3E	`	%60
#	%23	;	%3B
%	%25	/	%2F
{	%7B	?	%3F
}	%7D	:	%3A
	%7C	@	%40
\	%5C	=	%3D
^	%5E	&	%26
~	%7E	\$	%24

#### Structure:

<http://localhost:8890/CoCon/Voting/AddVotingTemplate/?Title={title}&NrOfOptions={nrOfOptions}&IsPrinted={isPrintedWhenStop}&IsVotingNumber={isVotingNumber}&AbstainOption={abstainOption}&AbstainIndex={abstainIndex}&Description1={description1}&Option1={option1}&SymbolIndex1={symbolIndex1}&LedColour1={ledColor1}&IsAbstain1={isAbstain1}&MsgOfMajority1={msgOfMajority1}&Description2={description2}&Option2={option2}&SymbolIndex2={symbolIndex2}&LedColour2={ledColor2}&IsAbstain2={isAbstain2}&MsgOfMajority2={msgOfMajority2}&Description3={description3}&Option3={option3}&SymbolIndex3={symbolIndex3}&LedColour3={ledColor3}&IsAbstain3={isAbstain3}&MsgOfMajority3={msgOfMajority3}&Description4={description4}&Option4={option4}&SymbolIndex4={symbolIndex4}&LedColour4={ledColor4}&IsAbstain4={isAbstain4}&MsgOfMajority4={msgOfMajority4}&Description5={description5}&Option5={option5}&SymbolIndex5={symbolIndex5}&LedColour5={ledColor5}&IsAbstain5={isAbstain5}&MsgOfMajority5={msgOfMajority5}&Description6={description6}&Option6={option6}&SymbolIndex6={symbolIndex6}&LedColour6={ledColor6}&IsAbstain6={isAbstain6}&MsgOfMajority6={msgOfMajority6}&Description7={description7}&Option7={option7}&SymbolIndex7={symbolIndex7}&LedColour7={ledColor7}&IsAbstain7={isAbstain7}&MsgOfMajority7={msgOfMajority7}&Description8={description8}&Option8={option8}&SymbolIndex8={symbolIndex8}&LedColour8={ledColor8}&IsAbstain8={isAbstain8}&MsgOfMajority8={msgOfMajority8}&Description9={description9}&Option9={option9}&SymbolIndex9={symbolIndex9}&LedColour9={ledColor9}&IsAbstain9={isAbstain9}&MsgOfMajority9={msgOfMajority9}&Description10={description10}&Option10={option10}&SymbolIndex10={symbolIndex10}&LedColour10={ledColor10}&IsAbstain10={isAbstain10}&MsgOfMajority10={msgOfMajority10}&BadgeOption={badgeOption}&OverallOption={overallOption}&IndividualOption={individualOption}&CanCorrect={canCorrect}&HasPin={hasPin}&IsWeightUsed={isWeightUsed}&>

[IsOperatorIndicated={isOperatorIndicated}&IsSignageIndicated={isSignageIndicated}&PanelIndicationOption={panelIndicationOption}&](#)  
[EnableAgendaTimer={isEnabledOfAgendaTimer}&DurationOfAgendaTimer={durationOfAgendaTimer}&WarningOfAgendaTimer={warningOfAgendaTimer}&CountDownOfAgendaTimer={isCountDownOfAgendaTimer}&ExceededOfAgendaTimer={canExceedOfAgendaTimer}&](#)  
[EnableVotingTimer={isEnabledOfVotingTimer}&DurationOfVotingTimer={durationOfVotingTimer}&WarningOfVotingTimer={warningOfVotingTimer}&CountDownOfVotingTimer={isCountDownOfVotingTimer}&ExceedOfVotingTimer={canExceedOfVotingTimer}&](#)  
[EnableQuorum={isEnabledOfQuorum}&AbsoluteOfQuorum={isAbsoluteOfQuorum}&ValueOfQuorum={valueOfQuorum}&OptionOfQuorum={optionOfQuorum}&EnableQuorumMsg={isEnabledOfQuorumMsg}&QuorumMsg={quorumMsg}&](#)  
[AbsoluteOfOutcome={isAbsoluteOfOutcome}&ValueOfMajority={valueOfMajority}&MajorityOption={majorityOption}&](#)  
[EnableOutcomMsg={isEnabledOfOutcomeMsg}&PassedMsg={passedMsg}&FailedMsg={failedMsg}&HungMsg={hungMsg}&EnabledMajorityMsg={isEnabledOfMajorityMsg}&](#)  
[HungVoteSetting={hungVoteSetting}&IsUnanimous={isUnanimous}](#)

**Parameters:**

Name	Description	Type
Title	Name of the Voting template and will keep the title due that agenda	string
NrOfOptions	Number of voting options	int
IsPrinted	Enabled the function for print after voting stop.	bool
IsVotingNumber	Enabled Symbol or Number for voting choice <b>Note: not applicable after 5.0 cocon</b>	bool
AbstainOption	Choose the Voting abstain Options Value :1 to 3	int
AbstainIndex	Select abstain index for move voting for Value : 1 to .. based on the voting options.	int
Description1	Description for voting option 1	string
Option1 (Option Color)	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. <b>Note: Make sure for correct RGB value.</b>	string
SymbolIndex1	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour1	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain1	Enabled the abstain function	bool
MsgOfMajority1	Message for majority	string
Description2	Description for voting option 2	string
Option2	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. <b>Note: Make sure for correct RGB value.</b>	string
SymbolIndex2	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour2	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain2	Enabled the abstain function	bool
MsgOfMajority2	Message for majority	string
Description3	Description for voting option 3	string
Option3	Voting Option Color Ex→%23FFFFFF00	string

	Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	
SymbolIndex3	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour3	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain3	Enabled the abstain function	bool
MsgOfMajority3	Message for majority	string
Description4	Description for voting option 4	string
Option4	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex4	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour4	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain4	Enabled the abstain function	bool
MsgOfMajority4	Message for majority	string
Description5	Description for voting option 5	string
Option5	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex5	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour5	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan	string
IsAbstain5	Enabled the abstain function	bool
MsgOfMajority5	Message for majority	string
Description6	Description for voting option 6	string
Option6	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex6	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour6	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain6	Enabled the abstain function	bool
MsgOfMajority6	Message for majority	string
Description7	Description for voting option 7	string
Option7	Voting Option Color Ex→%23FFFFFF00	string

	Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	
SymbolIndex7	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour7	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain7	Enabled the abstain function	bool
MsgOfMajority7	Message for majority	string
Description8	Description for voting option 8	string
Option8	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex8	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour8	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain8	Enabled the abstain function	bool
MsgOfMajority8	Message for majority	string
Description9	Description for voting option 9	string
Option9	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex9	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour9	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain9	Enabled the abstain function	bool
MsgOfMajority9	Message for majority	string
Description10	Description for voting option 10	string
Option10	Voting Option Color Ex→%23FFFFFF00 Where %23 is Prefix and 8 chars of the RGB value. Note: Make sure for correct RGB value.	string
SymbolIndex10	Buttons enabled for voting Index : 0 to 9 based on the options. Ex.	int
LedColour10	Voting LED color We support only 7 color now. Blue, White, Green, Yellow, Red, Magenta, Cyan <b>Note: LED color not support for more than 5 button options.</b>	string
IsAbstain10	Enabled the abstain function	bool



MsgOfMajority10	Message for majority	string
BadgeOption	Choose option for "Who can participate in voting" Value : 1 to 4 Ex. 1 = Only delegates with Badge 2 = Only delegates positioned on a seat 3 = Delegates with a badge or positioned on a seat. 4 = All voting units Note: badge options will be available based on the meeting type Note: Options is available based on the meeting login method.	int
OverallOption	Choose overall option "Who sees the overall results during the vote ?" Value : 1 to 4 1= no one 2 = Choose this option if the overall result should only be shown to the vote master. This is an option that can be selected in the Operator application. 3 = Choose this option if all Operator application should show the overall results during the vote. 4 = Choose this option if all operator & Signage application should show the overall results.	int
IndividualOption	Choose the option for "Who sees the individual voting results (during & after the vote)" Value: 1 to 5 1= No one ;Never 2 = No one during the vote 3 = Only vote master 4 = All operators applications 5 = Choose this option if everybody should see the individual results during the vote.	int
CanCorrect	Enabled the correction option Value : True means user can change voting	bool
HasPin	Enable the option for "Enter PinId for voting" Note: it will be used for IDC only.	bool
IsWeightUsed	Enable option to check the voting weight for voting result calculation.	bool
IsOperatorIndicated	Vote indication Enabled the Operator voting indication Note: it can configure based on the individual voting setting.	bool
IsSignageIndicated	Vote indication Enabled the Signage voting indication Note: it can configure based on the individual voting setting.	bool
PanelIndicationOption	Vote button indication Chose the vote panel indication option Value : 0 to 2 2 means "Real vote choice" Note: it can configure based on the individual voting setting.	int
EnableAgendaTimer	Enabled the timer for Agenda	bool
DurationOfAgendaTimer	Total Time for Agenda Ex. 00:05:00	String
WarningOfAgendaTimer	Warning time for the Agenda Ex. 00:01:00	String
CountDownOfAgendaTimer	Choose the agenda timer count-up / count-down	Bool
ExceedOfAgendaTimer	Allow exceed agenda timer	Bool
EnableVotingTimer	Enabled the timer for voting	Bool
DurationOfVotingTimer	Total Time for Agenda Ex. 00:05:00	String

WarningOfVotingTimer	Warning time for the Agenda Ex. 00:01:00	String
CountDownOfVotingTimer	Choose the agenda timer count-up / count-down	Bool
ExceedOfVotingTimer	Allow exceed voting timer	Bool
EnableQuorum	Enabled the Quorum Functionality 0 = Not enabled 1 = Enabled Default value is 0	Bool
AbsoluteOfQuorum	Choose Quorum define function for Absolute or relative Ex. True means Absolute enabled False means Relative option selected. (enabled)	Bool
ValueOfQuorum	Value of the Quorum	decimal
OptionOfQuorum	Choose the options for Presence of determination. Value : 1 to 5 1 = Introduced badges 2 = Presence detection phase before vote 3 = Delegates positioned on a seat 4 = Entered manually by operator 5 = Total number of votes	int
EnableQuorumMsg	Enabled the Quorum message option	bool
QuorumMsg	Message for Quorum	string
AbsoluteOfOutcome	Choose Majority define function for Absolute or relative Ex. True means Absolute enabled False means Relative option selected (enabled).	bool
ValueOfMajority	Value of the Majority	decimal
MajorityOption	Choose the majority options Value :1 to 2 1 = Number of delegates entitled to vote is used for majority calculation. 2 = Number of votes cast is used for majority calculation. Note: option 1 can only be selectable if the relative option is selected.	int
EnableOutcomMsg	Enabled Outcome message option.	bool
PassedMsg	Message for the Pass Majority	string
FailedMsg	Message for the Fail Majority	string
HungMsg	Message for the Hung Majority	string
EnabledMajorityMsg	Enabled Majority Achieved Details	bool
HungVoteSetting	Choose option for the hung vote solution Value : 1 to 5 Ex. 0 = Stop voting automatically. 1 = Keep the vote active until the Operator or Chairman manually stops it. 2 = Restart voting with all options available 3 = Determined by chairmen votes in this voting session. 4 = Stop voting and the operator decides. 5 = Determined by chairmen votes in a new chairmen-only voting session.	Int
IsUnanimous	If value is true, means Voting options for the voting session decision made by operator even though user can choose his own chose as well. (this option will used by the Cocon Op app only now) Note: default value is false	bool

**Example 1:**

<http://localhost:8890/CoCon/Voting/AddVotingTemplate/?Title=Test 223>

**Example 2:**

<http://localhost:8890/CoCon/Voting/AddVotingTemplate/?Title=TEST 201&NrOfOptions=5&IsPrinted=false&Description1=Pass&Option1=%230000FF&SymbolIndex1=0&LedColour1=Red&Description2=ABS&Option2=%23FFF000&SymbolIndex2=1&LedColour2=Red&Description3=Fail&Option3=%23FFFFFF00&SymbolIndex3=2&LedColour3=Blue&Description4=ISA&Option4=%2339FF14&SymbolIndex4=3&LedColour4=Yellow&Description5=XYZ&Option5=%23FF00FF&SymbolIndex5=4&LedColour5=Magenta&EnableQuorum=True&AbsoluteOfQuorum=False&OptionOfQuorum=1&EnableQuorumMsg=True&QuorumMsg=lt works&AbsoluteOfOutcome=False&ValueOfMajority=10&MajorityOption=1&EnableOutcomMsg=True&PassedMsg=Majority achieved&FailedMsg=fail&HungMsg=hung&EnabledMajorityMsg=True&HungVoteSetting=4&BadgeOption=1&EnableAgendaTimer=True&EnableVotingTimer=True&DurationOfAgendaTimer=00%3A05%3A00&WarningOfAgendaTimer=00%3A03%3A00&CountDownOfAgendaTimer=False&ExceedOfAgendaTimer=True&DurationOfVotingTimer=00%3A05%3A00&WarningOfVotingTimer=00%3A05%3A00&CountDownOfVotingTimer=False&ExceedOfVotingTimer=True&OverallOption=1&IndividualOption=1&CanCorrect=True&IsWeightUsed=True&IsOperatorIndicated=True&IsSignageIndicated=True&PanelIndicationOption=1>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**Update event for Example 1:**

```
{
  "VotingTemplateAdded": {
    "Id": 35,
    "Title": "Test 223",
    "NrOfOptions": 3,
    "IsPrinted": false,
    "IsVotingNumber": false,
    "BadgeOption": 4,
    "OverallOption": 4,
    "Options": [
      {
        "Id": 154,
        "Description": "",
        "Option": "#FF008000",
        "SymbolIndex": 1,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Accepted"
      },
      {
        "Id": 155,
        "Description": "",
        "Option": "#FFFFFF00",
        "SymbolIndex": 2,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Postponed"
      },
      {
        "Id": 156,
        "Description": "",
        "Option": "#FFFF0000",
        "SymbolIndex": 3,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Rejected"
      }
    ],
    "IndividualOption": 5,
    "CanCorrect": false,
    "HasPin": false,
    "IsWeightUsed": true,
    "IsOperatorIndicated": false,
    "IsSignageIndicated": false,
    "PanelIndicationOption": 0,
    "AgendaTimer": {
      "Id": 71,
      "EnableAgendaTimer": false,
      "DurationOfAgendaTimer": "00:00:00",
      "WarningOfAgendaTimer": "00:00:00",
      "CountDownOfAgendaTimer": true,
      "ExceedOfAgendaTimer": false
    },
    "VotingTimer": {
      "Id": 72,
      "EnableVotingTimer": false,
      "DurationOfVotingTimer": "00:00:00",
      "WarningOfVotingTimer": "00:00:00",
      "CountDownOfVotingTimer": true,
      "ExceedOfVotingTimer": false
    },
    "EnableQuorum": false,
    "AbsoluteOfQuorum": true,
    "ValueOfQuorum": 0.0,
    "OptionOfQuorum": 4,
    "EnableQuorumMsg": false,
    "QuorumMsg": "Quorum not OK - voting postponed",
    "AbsoluteOfOutcome": true,
    "ValueOfMajority": 1.0,
    "MajorityOption": 2,
    "EnableOutcomMsg": false,
    "PassedMsg": "Majority achieved",
    "FailedMsg": "Majority not achieved",
    "HungMsg": "Hung vote",
    "EnabledMajorityMsg": false,
    "HungVoteSetting": 1
  }
}
```

**Update event for Example 2:**

```
{
  "VotingTemplateAdded": {
    "Id": 33,
    "Title": "TEST 201",
    "NrOfOptions": 5,
    "IsPrinted": false,
    "IsVotingNumber": false,
    "BadgeOption": 1,
    "OverallOption": 1,
    "Options": [
      {
        "Id": 144,
        "Description": "Pass",
        "Option": "#0000FF",
        "SymbolIndex": 0,
        "LedColour": "Red",
        "IsAbstain": false,
        "MsgOfMajority": ""
      },
      {
        "Id": 145,
        "Description": "ABS",
        "Option": "#FFF000",
        "SymbolIndex": 1,
        "LedColour": "Red",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Accepted"
      },
      {
        "Id": 146,
        "Description": "Fail",
        "Option": "#FFFFFF00",
        "SymbolIndex": 2,
        "LedColour": "Blue",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Postponed"
      },
      {
        "Id": 147,
        "Description": "ISA",
        "Option": "#39FF14",
        "SymbolIndex": 3,
        "LedColour": "Yellow",
        "IsAbstain": false,
        "MsgOfMajority": "Motion Rejected"
      },
      {
        "Id": 148,
        "Description": "XYZ",
        "Option": "#FF00FF",
        "SymbolIndex": 4,
        "LedColour": "Magenta",
        "IsAbstain": false,
        "MsgOfMajority": ""
      }
    ],
    "IndividualOption": 1,
    "CanCorrect": true,
    "HasPin": false,
    "IsWeightUsed": true,
    "IsOperatorIndicated": true,
    "IsSignageIndicated": true,
    "PanelIndicationOption": 1,
    "AgendaTimer": {
      "Id": 67,
      "EnableAgendaTimer": true,
      "DurationOfAgendaTimer": "00:05:00",
      "WarningOfAgendaTimer": "00:03:00",
      "CountDownOfAgendaTimer": false,
      "ExceedOfAgendaTimer": true
    },
    "VotingTimer": {
      "Id": 68,
      "EnableVotingTimer": true,
      "DurationOfVotingTimer": "00:05:00",
      "WarningOfVotingTimer": "00:05:00",
      "CountDownOfVotingTimer": false,
      "ExceedOfVotingTimer": true
    },
    "EnableQuorum": true,
    "AbsoluteOfQuorum": false,
    "ValueOfQuorum": 0.0,
    "OptionOfQuorum": 1,
    "EnableQuorumM"
  }
}
```

sg":true,"QuorumMsg":"It works", "AbsoluteOfOutcome":false, "ValueOfMajority":10.0, "MajorityOption":1, "EnableOutcomMsg":true, "Passed Msg":"Majority achieved", "FailedMsg":"fail", "HungMsg":"hung", "EnabledMajorityMsg":true, "HungVoteSetting":4}}

### Example 3:

<http://localhost:8890/CoCon/Voting/AddVotingTemplate/?Title=TEST501&NrOfOptions=5&IsPrinted=false&Description1=Pass&Option1=%230000FF&SymbolIndex1=0&LedColour1=Red&Description2=ABS&Option2=%23FFFF0000&SymbolIndex2=1&LedColour2=Red&Description3=Fail&Option3=%23FFFFFF00&SymbolIndex3=2&LedColour3=Blue&Description4=ISA&Option4=%2339FF14&SymbolIndex4=3&LedColour4=Yellow&Description5=XYZ&Option5=%23FF00FF&SymbolIndex5=4&LedColour5=Magenta&EnableQuorum=True&AbsoluteOfQuorum=False&OptionOfQuorum=1&EnableQuorumMsg=True&QuorumMsg=It works&AbsoluteOfOutcome=False&ValueOfMajority=10&MajorityOption=1&EnableOutcomMsg=True&Passed Msg=Majority achieved&FailedMsg=fail&HungMsg=hung&EnabledMajorityMsg=True&HungVoteSetting=4&BadgeOption=1&EnableAgendaTimer=True&EnableVotingTimer=True&DurationOfAgendaTimer=00%3A05%3A00&WarningOfAgendaTimer=00%3A03%3A00&CountDownOfAgendaTimer=False&ExceedOfAgendaTimer=True&DurationOfVotingTimer=00%3A05%3A00&WarningOfVotingTimer=00%3A05%3A00&CountDownOfVotingTimer=False&ExceedOfVotingTimer=True&OverallOption=1&IndividualOption=1&CanCorrect=True&IsWeightUsed=True&IsOperatorIndicated=True&IsSignageIndicated=True&PanelIndicationOption=1&IsAbstain1=True&AbstainOption=3&AbstainIndex=2>

### Update event for Example 4 :

```
>>>{"VotingTemplateAdded":{"Id":36,"Title":"TEST501","NrOfOptions":5,"IsPrinted":false,"IsVotingNumber":false,"BadgeOption":1,"OverallOption":1,"Options":[{"Id":157,"Description":"Pass","Option":"#0000FF","SymbolIndex":0,"LedColour":"Red","IsAbstain":true,"MsgOfMajority":"","Id":158,"Description":"ABS","Option":"#FFFF0000","SymbolIndex":1,"LedColour":"Red","IsAbstain":false,"MsgOfMajority":"Motion Accepted"}, {"Id":159,"Description":"Fail","Option":"#FFFFFFF00","SymbolIndex":2,"LedColour":"Blue","IsAbstain":false,"MsgOfMajority":"Motion Postponed"}, {"Id":160,"Description":"ISA","Option":"#39FF14","SymbolIndex":3,"LedColour":"Yellow","IsAbstain":false,"MsgOfMajority":"Motion Rejected"}, {"Id":161,"Description":"XYZ","Option":"#FF00FF","SymbolIndex":4,"LedColour":"Magenta","IsAbstain":false,"MsgOfMajority":""}], "IndividualOption":1, "CanCorrect":true, "HasPin":false, "IsWeightUsed":true, "IsOperatorIndicated":true, "IsSignageIndicated":true, "PanelIndicationOption":1, "AgendaTimer":{"Id":73, "EnableAgendaTimer":true, "DurationOfAgendaTimer":"00:05:00", "WarningOfAgendaTimer":"00:03:00", "CountDownOfAgendaTimer":false, "ExceedOfAgendaTimer":true}, "VotingTimer":{"Id":74, "EnableVotingTimer":true, "DurationOfVotingTimer":"00:05:00", "WarningOfVotingTimer":"00:05:00", "CountDownOfVotingTimer":false, "ExceedOfVotingTimer":true}, "EnableQuorum":true, "AbsoluteOfQuorum":false, "ValueOfQuorum":0.0, "OptionOfQuorum":1, "EnableQuorumMsg":true, "QuorumMsg":"It works", "AbsoluteOfOutcome":false, "ValueOfMajority":10.0, "MajorityOption":1, "EnableOutcomMsg":true, "Passed Msg":"Majority achieved", "FailedMsg":"fail", "HungMsg":"hung", "EnabledMajorityMsg":true, "HungVoteSetting":4}}
```

### Update event for Example 5 : With additional para "IsUnanimous"

```
{"VotingTemplateAdded":{"Id":3,"Title":"Test223","NrOfOptions":3,"IsPrinted":false,"IsVotingNumber":false,"BadgeOption":4,"OverallOption":4,"Options":[{"Id":66,"Description":"","Option":"#FF008000","SymbolIndex":1,"LedColour":"Blue","IsAbstain":false,"MsgOfMajority":"Motion Accepted"}, {"Id":67,"Description":"","Option":"#FFFFFFF00","SymbolIndex":2,"LedColour":"Blue","IsAbstain":false,"MsgOfMajority":"Motion Postponed"}, {"Id":68,"Description":"","Option":"#FFFF0000","SymbolIndex":3,"LedColour":"Blue","IsAbstain":false,"MsgOfMajority":"Motion Rejected"}], "IndividualOption":5, "CanCorrect":false, "HasPin":false, "IsWeightUsed":true, "IsOperatorIndicated":false, "IsSignageIndicated":false, "PanelIndicationOption":0, "AgendaTimer":{"Id":109, "EnableAgendaTimer":false, "DurationOfAgendaTimer":"00:00:00", "WarningOfAgendaTimer":"00:00:00", "CountDownOfAgendaTimer":true, "ExceedOfAgendaTimer":false}, "VotingTimer":{"Id":110, "EnableVotingTimer":false, "DurationOfVotingTimer":"00:00:00", "WarningOfVotingTimer":"00:00:00", "CountDownOfVotingTimer":true, "ExceedOfVotingTimer":false}, "EnableQuorum":false, "AbsoluteOfQuorum":true, "ValueOfQuorum":0.0, "OptionOfQuorum":4, "EnableQuorumMsg":false, "QuorumMsg":"Quorum not OK - voting"}}
```

postponed", "AbsoluteOfOutcome":true, "ValueOfMajority":1.0, "MajorityOption":2, "EnableOutcomMsg":false, "PassedMsg": "Majority achieved", "FailedMsg": "Majority not achieved", "HungMsg": "Hung vote", "EnabledMajorityMsg":false, "HungVoteSetting":1, "IsUnanimous":false}}

### 3.3.5.4 GetVotingState

API Module	Cocon for Plixus Core	Plixus Core
GetVotingState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gets the voting state of currently active agenda item. Only applicable for voting items. If the currently active agenda item is not a voting agenda item, then the state Stop will be returned

**Structure:**

<http://localhost:8890/CoCon/Voting/GetVotingState>

**Parameters:**

Name	Description	Type
State	The state of voting Start Stop Pause (Idle)Clear Retrieve	string

**Example:**

<http://localhost:8890/CoCon/Voting/GetVotingState>

**Return:**

```
{"GetVotingState":{"State":"VotingIdle"}}
```

### 3.3.5.5 GetAmountOfVotes

API Module	Cocon for Plixus Core	Plixus Core
GetAmountOfVotes	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Will help user to get the current vote amount during the voting. (voting is running / paused or stopped)  
Note: if voting state is idle then it will reply with no voting is running.

**Structure:**

<http://localhost:8890/CoCon/Voting/GetAmountOfVotes>

**Example:**

<http://localhost:8890/CoCon/Voting/GetAmountOfVotes>

**Return:**

```
{"GetAmountOfVotes":{"Total voted":0,"Total not voted":15,"Entitled to vote":15}}
```

If voting is not running :

There is no running voting

### 3.3.5.6 GetGeneralVotingResults

API Module	Cocon for Plixus Core	Plixus Core
GetGeneralVotingResults	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Will help user to get the overall voting result for the active voting agenda after the voting stop.

**Structure:**

<http://localhost:8890/CoCon/Voting/GetGeneralVotingResults/?Id={id}>

**Example:**

<http://localhost:8890/CoCon/Voting/GetGeneralVotingResults/?Id=20>

**Return:**

```
{"GetGeneralVotingResults":{"Id":100,"VotingResults":{"Total":{"Count":4,"Weight":4.0},"Voted":{"Count":1,"Weight":1.0},"NotVoted":{"Count":3,"Weight":3.0},"Options":[{"Id":181,"Name":"","Color":"#FF008000","Votes":{"Count":0,"Weight":0.0}},{"Id":182,"Name":"","Color":"#FFFFFF00","Votes":{"Count":0,"Weight":0.0}},{"Id":183,"Name":"","Color":"#FFFF0000","Votes":{"Count":1,"Weight":1.0}}],"AuthorityAssigned":{"Present":0,"Voted":0,"Register":0}}}}
```

1 : if agenda is not voting agenda or voting agenda does not have result.

-1 : internal exception.

### 3.3.6 Timer

#### 3.3.6.1 Get

API Module	Cocon for Plixus Core	Plixus Core
Get	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gets an update of all timers

**Structure:**

<http://localhost:8890/CoCon/Timer/Get>

**Return:**

```
{"GetAllTimers":{"MeetingTimer":{"Timer":{"TotalTime":"02:00:00","TimeUsed":"00:31:23","CountingDown":true,"WarningTime":"00:05:00"},"ActiveAgendaItemTimer":{"Timer":{"TotalTime":"02:00:00","TimeUsed":"00:31:23","CountingDown":true,"WarningTime":"00:05:00"},"VotingTimer":{"Timer":null},"SeatsTimer":[{"SeatNr":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}],"DelegatesTimer":[{"DelegatId":10,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}],"GroupsTimer":[{"GroupId":1,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}},{"GroupId":11,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}]}}
```

#### 3.3.6.2 GetDelegateTimers

API Module	Cocon for Plixus Core	Plixus Core
GetDelegateTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all delegates timer. (meeting users)

Note: Delegate on seat only can have timer.

**Structure:**

<http://localhost:8890/CoCon/Timer/GetDelegateTimers>

**Return:**

```
{"GetDelegateTimers":[{"DelegatId":2,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}},{"DelegatId":3,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}},{"DelegatId":4,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}]}
```

### 3.3.6.3 GetSeatTimers

API Module	Cocon for Plixus Core	Plixus Core
GetSeatTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all seat timers. (empty seat without delegate on seat.)

**Structure:**

<http://localhost:8890/CoCon/Timer/GetSeatTimers>

**Return:**

```
{"GetSeatTimers":[{"SeatNr":4,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}},{"SeatNr":5,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}]}
```

### 3.3.6.4 GetGroupTimers

API Module	Cocon for Plixus Core	Plixus Core
GetGroupTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all groups timer. (Meeting groups timer only)

**Structure:**

<http://localhost:8890/CoCon/Timer/GetGroupTimers>

**Return:**

```
{"GetGroupTimers":[{"GroupId":3,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}},{"GroupId":4,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:00","CountingDown":true,"WarningTime":"00:01:00"}]}
```

### 3.3.6.5 GetActiveTimers

API Module	Cocon for Plixus Core	Plixus Core
GetActiveTimers	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all active timers info. (Seat timer; Delegate timer ; Group timer )

**Structure:**

<http://localhost:8890/CoCon/Timer/GetActiveTimers>

**Return:**

```
{"GetActiveTimers":{"SeatsTimer":[{"SeatNr":8,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:15","CountingDown":true,"WarningTime":"00:01:00"}],"DelegatesTimer":[{"DelegatId":43,"Timer":{"TotalTime":"00:05:00","TimeUsed":"00:00:17","CountingDown":true,"WarningTime":"00:01:00"}],"GroupsTimer":[]}}
```

### 3.3.6.6 SetDelegateTimer

API Module	Cocon for Plixus Core	Plixus Core
SetDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set delegate timer.

Note: Delegate timer will enabled auto and set timer, if timer was not enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/SetDelegateTimer/?DelegateId={delegateId}&EndTime={endTime}&WarningTime={warningTime}&CountDown={countDown}&IncludeChairman={includeChairman}&SpeechTimeCanExceed={speechTimeCanExceed}&AutoSwitchOffMic={autoSwitchOffMic}&ResetWhenMicOff={resetWhenMicOff}&Enable={enable}>

**Parameters:**

Name	Description	Type
DelegateId	The Delegate id from DB	int
EndTime	End time for the timer	string
WarningTime	Warning time for the timer	string
CountDown	Counting up or down for timer	bool
IncludeChairman	Include chairman	bool
SpeechTimeCanExceed	Speech time can exceed (high priority)	bool
AutoSwitchOffMic	Auto switch off mic if time is over (Low Priority)	bool
ResetWhenMicOff	Reset timer once mic off.	bool
Enable	Set enabled / disabled timer	bool

Note: if SpeechTimeCanExceed =True and AutoSwitchOffMic = True, then SpeechTimeCanExceed has higher priority than AutoSwitchOffMic.

**Example:**

<http://localhost:8890/CoCon/Timer/SetDelegateTimer/?DelegateId=2&EndTime=00:01:00&WarningTime=00:00:30&CountDown=True&IncludeChairman=True&SpeechTimeCanExceed=True&AutoSwitchOffMic=True&ResetWhenMicOff=True&Enable=True>

**Return: string**

```
{"SetDelegateTimer":{"DelegateId":71,"Endtime":"00:01:00","WarningTime":"00:00:30","CountDown":true,"IncludeChairman":true,"SpeechTimeCanExceed":true,"AutoSwitchOffMic":false,"ResetWhenMicOff":false,"Enable":true}}
```

### 3.3.6.7 SetGroupTimer

API Module	Cocon for Plixus Core	Plixus Core
SetGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set group timer.

Note: Delegate timer will enabled auto and set timer, if timer was not enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/SetGroupTimer/?GroupId={groupId}&EndTime={endTime}&WarningTime={warningTime}&CountDown={countDown}&IncludeChairman={includeChairman}&SpeechTimeCanExceed={speechTimeCanExceed}&AutoSwitchOffMic={autoSwitchOffMic}&CountForEverySpeaker={countForEverySpeaker}&Enable={enable}>

**Parameters:**

Name	Description	Type
GroupId	The Group id from DB	int



EndTime	End time for the timer	string
WarningTime	Warning time for the timer	string
CountDown	Counting up or down for timer	bool
IncludeChairman	Include chairman	bool
SpeechTimeCanExceed	Speech time can exceed (high priority)	bool
AutoSwitchOffMic	Auto switch off mic if time is over (Low Priority)	bool
CountForEverySpeaker	Count for every speaker belong to that group	bool
Enable	Set enabled / disabled timer	bool

Note: if SpeechTimeCanExceed = True and AutoSwitchOffMic = True, then SpeechTimeCanExceed has higher priority than AutoSwitchOffMic.

**Example:**

<http://localhost:8890/CoCon/Timer/SetGroupTimer/?GroupId=1&EndTime=00:10:00&WarningTime=00:01:30&CountDown=True&IncludeChairman=True&SpeechTimeCanExceed=True&AutoSwitchOffMic=True&CountForEverySpeaker=True&Enable=True>

**Return: string**

```
{"SetGroupTimer":{"GroupId":9,"Endtime":"00:03:00","WarningTime":"00:01:30","CountDown":true,"IncludeChairman":true,"SpeechTimeCanExceed":true,"AutoSwitchOffMic":false,"CountForEverySpeaker":true,"Enable":true}}
```

**3.3.6.8 SetGlobalDelegateTimer**

API Module	Cocon for Plixus Core	Plixus Core
SetGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set global delegate timer. (for all meeting user )

Note: Delegate timer will enabled auto and set timer, if timer was not enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/SetGlobalDelegateTimer/?EndTime={endTime}&WarningTime={warningTime}&CountDown={countDown}&IncludeChairman={includeChairman}&SpeechTimeCanExceed={speechTimeCanExceed}&AutoSwitchOffMic={autoSwitchOffMic}&ResetWhenMicOff={resetWhenMicOff}&Enable={enable}>

**Parameters:**

Name	Description	Type
EndTime	End time for the timer	string
WarningTime	Warning time for the timer	string
CountDown	Counting up or down for timer	bool
IncludeChairman	Include chairman	bool
SpeechTimeCanExceed	Speech time can exceed (high priority)	bool
AutoSwitchOffMic	Auto switch off mic if time is over (Low Priority)	bool
ResetWhenMicOff	Reset timer once mic off.	bool
Enable	Set enabled / disabled timer	bool

Note: if SpeechTimeCanExceed = True and AutoSwitchOffMic = True, then SpeechTimeCanExceed has higher priority than AutoSwitchOffMic.

**Example:**

<http://localhost:8890/CoCon/Timer/SetGlobalDelegateTimer/?EndTime=00:02:00&WarningTime=00:00:30&CountDown=False&IncludeChairman=True&SpeechTimeCanExceed=True&AutoSwitchOffMic=True&ResetWhenMicOff=True&Enable=True>

**Return: string**

```
{"SetGlobalDelegateTimer":{"EndTime":"00:02:00","WarningTime":"00:00:30","CountDown":false,"IncludeChairman":true,"SpeechTimeCanExceed":true,"AutoSwitchOffMic":false,"ResetWhenMicOff":true}}
```

### 3.3.6.9 SetGlobalSeatTimer

API Module	Cocon for Plixus Core	Plixus Core
SetGlobalSeatTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set global seat timer. (For all empty seat in running meeting.)

Note: Delegate timer will enabled auto and set timer, if timer was not enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/SetGlobalSeatTimer/?EndTime={endTime}&WarningTime={warningTime}&CountDown={countDown}&IncludeChairman={includeChairman}&SpeechTimeCanExceed={speechTimeCanExceed}&AutoSwitchOffMic={autoSwitchOffMic}&ResetWhenMicOff={resetWhenMicOff}&Enable={enable}>

**Parameters:**

Name	Description	Type
EndTime	End time for the timer	string
WarningTime	Warning time for the timer	string
CountDown	Counting up or down for timer	bool
IncludeChairman	Include chairman	bool
SpeechTimeCanExceed	Speech time can exceed (high priority)	bool
AutoSwitchOffMic	Auto switch off mic if time is over (Low Priority)	bool
ResetWhenMicOff	Reset timer once mic off.	bool
Enable	Set enabled / disabled timer	bool

Note: if SpeechTimeCanExceed =True and AutoSwitchOffMic = True, then SpeechTimeCanExceed has higher priority than AutoSwitchOffMic.

**Example:**

<http://localhost:8890/CoCon/Timer/SetGlobalSeatTimer/?EndTime=00:02:00&WarningTime=00:00:30&CountDown=True&IncludeChairman=True&SpeechTimeCanExceed=True&AutoSwitchOffMic=True&ResetWhenMicOff=True&Enable=True>

**Return: string**

```
{"SetGlobalSeatTimer":{"EndTime":"00:01:00","WarningTime":"00:00:30","CountDown":true,"IncludeChairman":true,"SpeechTimeCanExceed":true,"AutoSwitchOffMic":false,"ResetWhenMicOff":true}}
```

### 3.3.6.10 SetGlobalGroupTimer

API Module	Cocon for Plixus Core	Plixus Core
SetGlobalSeatTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set global group timer. (For all running meeting groups.)

Note: Delegate timer will enabled auto and set timer, if timer was not enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/SetGlobalGroupTimer/?EndTime={endTime}&WarningTime={warningTime}&CountDown={countDown}&IncludeChairman={includeChairman}&SpeechTimeCanExceed={speechTimeCanExceed}&AutoSwitchOffMic={autoSwitchOffMic}&CountForEverySpeaker={countForEverySpeaker}&Enable={enable}>

**Parameters:**

Name	Description	Type
EndTime	End time for the timer	string
WarningTime	Warning time for the timer	string
CountDown	Counting up or down for timer	bool
IncludeChairman	Include chairman	bool
SpeechTimeCanExceed	Speech time can exceed (high priority)	bool
AutoSwitchOffMic	Auto switch off mic if time is over (Low Priority)	bool
CountForEverySpeaker	Count for every speaker belong to that group	bool
Enable	Set enabled / disabled timer	bool

Note: if SpeechTimeCanExceed =True and AutoSwitchOffMic = True, then SpeechTimeCanExceed has higher priority than AutoSwitchOffMic.

**Example:**

<http://localhost:8890/CoCon/Timer/SetGlobalGroupTimer/?EndTime=00:01:00&WarningTime=00:00:30&CountDown=True&IncludeChairman=True&SpeechTimeCanExceed=True&AutoSwitchOffMic=True&CountForEverySpeaker=False&Enable=True>

**Return: string**

```
{"SetGlobalGroupTimer":{"Endtime":"00:01:00","WarningTime":"00:00:30","CountDown":true,"IncludeChairman":true,"SpeechTimeCanExceed":true,"AutoSwitchOffMic":false,"CountForEverySpeaker":false}}
```

### 3.3.6.11 ResetDelegateTimer

API Module	Cocon for Plixus Core	Plixus Core
ResetDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Reset delegate timer to the configured time if user have running or used timer in active meeting.

Note: Delegate should be a meeting user and his timer should be enabled

**Structure:**

<http://localhost:8890/CoCon/Timer/ResetDelegateTimer/?DelegateId={id}>

**Parameters:**

Name	Description	Type
DelegateId	Id fo the delegate	string

**Example:**

<http://192.168.8.165:8890/CoCon/Timer/ResetDelegateTimer/?DelegateId=10>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.6.12 ResetGlobalDelegateTimer

API Module	Cocon for Plixus Core	Plixus Core
ResetGlobalDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Reset global delegate timer to the configured time for the all meeting user whos timer is enabled in active meeting.

**Structure:**

<http://localhost:8890/CoCon/Timer/ResetGlobalDelegateTimer>

**Example:**

<http://localhost:8890/CoCon/Timer/ResetGlobalDelegateTimer>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.6.13 ResetGroupTimer

API Module	Cocon for Plixus Core	Plixus Core
ResetGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Reset group timer to the configured time if the group has running or used timer in active meeting.  
Note: Group should be a meeting Group and his timer should be enabled

**Structure:**

<http://localhost:8890/CoCon/Timer/ResetDelegateTimer/?DelegateId={id}>

**Parameters:**

Name	Description	Type
GroupId	Id fo the meeting group	string

**Example:**

<http://192.168.8.165:8890/CoCon/Timer/ResetGroupTimer/?GroupId=1>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.6.14 ResetGlobalGroupTimer

API Module	Cocon for Plixus Core	Plixus Core
ResetGlobalGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Reset global group timer to the configured time for the all meeting Group whos timer is enabled in active meeting.

**Structure:**

<http://192.168.8.165:8890/CoCon/Timer/ResetGlobalGroupTimer>

**Example:**

<http://192.168.8.165:8890/CoCon/Timer/ResetGlobalGroupTimer>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.6.15 ChangeDelegateTimer

API Module	Cocon for Plixus Core	Plixus Core
ChangeDelegateTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Change delegate timer to the configured time if user have running or used timer in active meeting.

Note: Delegate should be a meeting user and his timer should be enabled.

**Structure:**

<http://localhost:8890/CoCon/Timer/ChangeDelegateTimer/?DelegateId={delegateId}&UsedTime={usedTime}>

**Parameters:**

Name	Description	Type
DelegateId	Id fo the delegate	String
UsedTime	Time which user would like to update	String

**Example:**

<http://192.168.8.165:8890/CoCon/Timer/ChangeDelegateTimer/?DelegateId=9&UsedTime=00:00:30>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.6.16 ChangeGroupTimer

API Module	Cocon for Plixus Core	Plixus Core
ChangeGroupTimer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Change group timer to the configured time if the group has running or used timer in active meeting.

Note: Group should be a meeting Group and his timer should be enabled

**Structure:**

<http://localhost:8890/CoCon/Timer/ChangeGroupTimer/?GroupId={groupId}&UsedTime={usedTime}>

**Parameters:**

Name	Description	Type
GroupId	Id fo the meeting group	string

**Example:**

<http://192.168.8.165:8890/CoCon/Timer/ChangeGroupTimer/?GroupId=1&UsedTime=00:01:00>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.7 Delegate

#### 3.3.7.1 GetDelegatesInMeeting

API Module	Cocon for Plixus Core	Plixus Core
GetDelegatesInMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all the delegates information in the special meeting

**Structure:**

<http://localhost:8890/CoCon/Delegate/GetDelegatesInMeeting/?MeetingId=<MeetingId>>

**Parameters:**

Name	Description	Type
MeetingId	The id of the meeting those delegates in	int

**Example:**

<http://localhost:8890/CoCon/Delegate/GetDelegatesInMeeting/?MeetingId=8>

**Return:**

```
{
  "GetDelegatesInMeeting": {
    "Delegates": [
      {
        "Id": 240,
        "FirstName": "",
        "Name": "AUCOUTURIER",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 9,
            "Name": "G9"
          }
        ],
        "SeatNumber": 2,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 274,
        "FirstName": "",
        "Name": "AUDIBERT8TROIN",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 10,
            "Name": "G10"
          }
        ],
        "SeatNumber": 3,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 275,
        "FirstName": "",
        "Name": "BARREILLE",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 10,
            "Name": "G10"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 261,
        "FirstName": "",
        "Name": "BATTIN",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "Mr",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 10,
            "Name": "G10"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 418,
        "FirstName": "David",
        "Name": "Balmer",
        "MiddleName": null,
        "Street": "Les Gasses 27",
        "StreetNumber": 0,
        "PostCode": "2946",
        "City": "Miécourt",
        "Country": null,
        "Title": "Député",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 18,
            "Name": "PLR"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 348,
        "FirstName": "D.",
        "Name": "BATAILLE",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 14,
            "Name": "GROUPE PS"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 336,
        "FirstName": "B.",
        "Name": "BAUDOUX",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 13,
            "Name": "GROUPE PC"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 334,
        "FirstName": "C.",
        "Name": "BEAUCHAMP",
        "MiddleName": null,
        "Street": "",
        "StreetNumber": 0,
        "PostCode": "",
        "City": "",
        "Country": null,
        "Title": "",
        "BirthDate": "",
        "District": null,
        "Biography": "",
        "Groups": [
          {
            "Id": 13,
            "Name": "GROUPE PC"
          }
        ],
        "SeatNumber": null,
        "BadgeNumber": null,
        "FingerprinterData": "",
        "KeypadLoginCode": ""
      },
      {
        "Id": 457,
        "FirstName": "Elisabeth",
        "Name": "Baume-Schneider",
        "MiddleName": null,
        "Street": "Rue du 24-Septembre 2",
        "StreetNumber": 0,
        "PostCode": "2800",
        "City": "Delémont",
        "Country": "",
        "Title": "Ministre",
        "BirthDate": "",
        "District": ""
      }
    ]
  }
}
```

```
,"Biography":"","Groups":[{"Id":20,"Name":"PS"}],"SeatNumber":null,"BadgeNumber":"A4F538DE","FingerprinterData":"","KeypadLoginCode":""}]}}
```

### 3.3.7.2 GetDelegatesInCurrentMeeting

API Module	Cocon for Plixus Core	Plixus Core
GetDelegatesInCurrentMeeting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all the delegates information in the current running meeting

**Structure:**

<http://localhost:8890/CoCon/Delegate/GetDelegatesInCurrentMeeting>

**Return:**

```
{"GetDelegatesInCurrentMeeting":{"IsMeetingRunning":true,"Delegates":[{"Id":240,"FirstName":"","Name":"AUC OUTURIER","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":9,"Name":"G9"}],"SeatNumber":2,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:274,"FirstName":"","Name":"AUDIBERT8TROIN","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":10,"Name":"G10"}],"SeatNumber":3,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:275,"FirstName":"","Name":"BARREILLE","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":10,"Name":"G10"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:261,"FirstName":"","Name":"BATTIN","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"Mr","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":10,"Name":"G10"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:418,"FirstName":"David","Name":"Balmer","MiddleName":null,"Street":"Les Gasses 27","StreetNumber":0,"PostCode":"2946","City":"Miécourt","Country":null,"Title":"Député","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":18,"Name":"PLR"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:348,"FirstName":"D.","Name":"BATAILLE","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":14,"Name":"GROUPE PS"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:336,"FirstName":"B.","Name":"BAUDOUX","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":13,"Name":"GROUPE PC"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:334,"FirstName":"C.","Name":"BEAUCHAMP","MiddleName":null,"Street":"","StreetNumber":0,"PostCode":"","City":"","Country":null,"Title":"","BirthDate":"","District":null,"Biography":"","Groups":[{"Id":13,"Name":"GROUPE PC"}],"SeatNumber":null,"BadgeNumber":null,"FingerprinterData":"","KeypadLoginCode":""},{Id:457,"FirstName":"Élisabeth","Name":"Baume-Schneider","MiddleName":null,"Street":"Rue du 24-Septembre 2","StreetNumber":0,"PostCode":"2800","City":"Delémont","Country":"","Title":"Ministre","BirthDate":"","District":"","Biography":"","Groups":[{"Id":20,"Name":"PS"}],"SeatNumber":null,"BadgeNumber":"A4F538DE","FingerprinterData":"","KeypadLoginCode":""}]}}
```

### 3.3.7.3 ClearAllDelegatesAndGroups

API Module	Cocon for Plixus Core	Plixus Core
ClearAllDelegatesAndGroups	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Clear all the delegates and groups in the database.

**Structure:**

<http://localhost:8890/CoCon/Delegate/ClearAllDelegatesAndGroups>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

#### Update event:

```
{"DelegateRemoved":[2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,144]}
```

#### 3.3.7.4 GetAllDelegates

API Module	Cocon for Plixus Core	Plixus Core
GetAllDelegates	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get all delegates list from the DB

#### Structure:

<http://localhost:8890/CoCon/Delegate/GetAllDelegates>

#### Example:

<http://localhost:8890/CoCon/Delegate/GetAllDelegates>

#### Return:

```
>>>{"GetAllDelegates":{"Delegates":[{"Id":81,"Name":"Vande Lanotte","MiddleName":null,"FirstName":"Johan","Title":"Mr","BadgeNumber":null,"UserName":null,"Password":null,"Street":"","StreetNumber":"","City":"","PostCode":"","Country":"","District":"","BirthDate":"","Email":null,"PhoneNr":null,"VotingRight":true,"VotingWeight":1,"FingerprinterData":"","KeypadLoginCode":"","Id":101,"Name":"Aers","MiddleName":null,"FirstName":"Wilfried","Title":"Mr","BadgeNumber":"64CBA3DB","UserName":"","Password":null,"Street":"","StreetNumber":"","City":"","PostCode":"","Country":"","District":"","BirthDate":"","Email":"","PhoneNr":"","VotingRight":true,"VotingWeight":1,"FingerprinterData":"","KeypadLoginCode":"","Id":372,"Name":"66","MiddleName":null,"FirstName":"","Title":"","BadgeNumber":null,"UserName":null,"Password":null,"Street":"","StreetNumber":"","City":"","PostCode":"","Country":null,"District":null,"BirthDate":"","Email":null,"PhoneNr":null,"VotingRight":true,"VotingWeight":1,"FingerprinterData":"","KeypadLoginCode":"","Id":463,"Name":"Ackermann","MiddleName":null,"FirstName":"Marcel","Title":"Député","BadgeNumber":"14EFB4E2","UserName":"","Password":null,"Street":"Route Principale 11c","StreetNumber":"","City":"Bourrignon","PostCode":"2803","Country":"","District":"","BirthDate":"","Email":"","PhoneNr":"","VotingRight":true,"VotingWeight":1,"FingerprinterData":"","KeypadLoginCode":"","Id":532,"Name":"76","MiddleName":null,"FirstName":"76","Title":"","BadgeNumber":null,"UserName":null,"Password":null,"Street":"","StreetNumber":"","City":"","PostCode":"","Country":null,"District":null,"BirthDate":"","Email":null,"PhoneNr":null,"VotingRight":true,"VotingWeight":1,"FingerprinterData":"","KeypadLoginCode":""}]}}
```

#### 3.3.7.5 AddDelegate

API Module	Cocon for Plixus Core	Plixus Core
AddDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add delegate to the DB.

#### Structure:

<http://localhost:8890/CoCon/Delegate/AddDelegate/?Name={name}&FirstName={firstName}&Title={title}&MiddleName={middleName}&BadgeNr={badgeNr}&UserName={userName}&Password={password}&Street={street}&StreetNr={streetNr}&City={city}&PostalCode={postalCode}&Country={country}&District={district}&BirthDate={birthDate}&Email={email}&PhoneNr={phoneNr}&Biography={biography}&VotingRight={votingRight}&VotingWeight={votingWeight}&PinId={pinId}&FingerprinterData={fingerprinterData}&KeypadLoginCode={keypadLoginCode}>



Name	Description	Type
Name	The name of the delegate	string
FirstName	The first name of the delegate	string
Title	The title of the delegate [Optional parameter]	string
MiddleName	MiddleName of the delegate [Optional parameter]	string
BadgeNr	The badge number [Optional parameter]	int
UserName	The username for the delegate [Optional parameter]	string
Password	The password for the delegate [Optional parameter]	string
Street	The street for the delegate [Optional parameter]	string
StreetNr	The street number for the delegate [Optional parameter]	string
City	The city for the delegate [Optional parameter]	string
PostalCode	The postal code for the delegate [Optional parameter]	string
Country	The country for the delegate [Optional parameter]	string
District	The district for the delegate [Optional parameter]	string
BirthDate	The delegate's birthday [Optional parameter]	string
Email	The delegate's email address [Optional parameter]	string
PhoneNr	The delegate's phone number [Optional parameter]	String
Biography	Biography of the delegate [Optional parameter]	string
VotingRight	Does the delegate have the right to vote? Default value : True [Optional parameter]	Bool
VotingWeight	The voting weight to apply for this delegate Default value : 1 [Optional parameter]	Int
PinId	PinId to login for voting Must be 4 number only (0-9) [Optional parameter]	Int
FingerprinterData	Fingerprint data to open the Unicos IDC screen [Optional parameter]	string
KeypadLoginCode	Code to login into the Unicos FMM (IDC) [Optional parameter]	int

**Example:**

<http://localhost:8890/CoCon/Delegate/AddDelegate/?Name=Isha&FirstName=Sakhalakr&Title=Mrs.>  
<http://localhost:8890/CoCon/Delegate/AddDelegate/?Name=Isha&FirstName=Sakhalakr&Title=Mrs.&PinId=1234>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**Update event :**

```
{"DelegateAdded":{"Delegateld":534,"Name":"Isha","MiddleName":null,"FirstName":"Sakhalakr","Title":"Mrs.,"
BadgeNr":"","UserName":"","Password":"","Street":"","StreetNr":0,"PostalCode":"","City":"","Country":"","Birth
Date":"","District":"","Email":"","PhoneNr":"","VotingRight":true,"VotingWeight":1,"PinId":"","FingerprinterData":"","
KeypadLoginCode":""}}
```

### 3.3.7.6 RemoveDelegate

API Module	Cocon for Plixus Core	Plixus Core
RemoveDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Remove delegate from the DB.

**Structure:**

<http://localhost:8890/CoCon/Delegate/RemoveDelegate/?DelegateId={delegateId}>

**Example:**

<http://localhost:8890/CoCon/Delegate/RemoveDelegate/?DelegateId=40>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**Update Event :**

{"DelegateRemoved":[40]}

### 3.3.7.7 EditDelegate

API Module	Cocon for Plixus Core	Plixus Core
EditDelegate	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Edit exiting delegate in DB.

**Structure:**

<http://localhost:8890/CoCon/Delegate/EditDelegate/?DelegateId={delegateId}&Name={name}&FirstName={firstName}&Title={title}&MiddleName={middleName}&BadgeNr={badgeNr}&UserName={userName}&Password={password}&Street={street}&StreetNr={streetNr}&City={city}&PostalCode={postalCode}&Country={country}&District={district}&BirthDate={birthDate}&Email={email}&PhoneNr={phoneNr}&Biography={biography}&VotingRight={votingRight}&VotingWeight={votingWeight}&PinId={pinId}&FingerprinterData={fingerprinterData}&KeypadLoginCode={keypadLoginCode}>

Name	Description	Type
DelegateId	The Id of the delegate	int
Name	The name of the delegate	string
FirstName	The first name of the delegate	string
Title	The title of the delegate [Optional parameter]	string
MiddleName	MiddleName of the delegate [Optional parameter]	string
BadgeNr	The badge number [Optional parameter]	int
UserName	The username for the delegate [Optional parameter]	string
Password	The password for the delegate [Optional parameter]	string
Street	The street for the delegate [Optional parameter]	string
StreetNr	The street number for the delegate [Optional parameter]	string
City	The city for the delegate [Optional parameter]	string
PostalCode	The postal code for the delegate [Optional parameter]	string

Country	The country for the delegate [Optional parameter]	string
District	The district for the delegate [Optional parameter]	string
BirthDate	The delegate's birthday [Optional parameter]	string
Email	The delegate's email address [Optional parameter]	string
PhoneNr	The delegate's phone number [Optional parameter]	String
Biography	Biography of the delegate [Optional parameter]	string
VotingRight	Does the delegate have the right to vote? [Optional parameter]	Bool
VotingWeight	The voting weight to apply for this delegate [Optional parameter]	Int
PinId	PinId to login for voting Must be 4 number only (0-9) [Optional parameter]	Int
FingerPrinterData	Fingerprint data to open the Unicos IDC screen [Optional parameter]	string
KeypadLoginCode	Code to login into the Unicos FMM (IDC) [Optional parameter]	int

**Example:**

<http://localhost:8890/CoCon/Delegate/EditDelegate/?DelegateId=505&Name=Manoj&FirstName=Patel&Title=Mr.&PinId=9876>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**Update Event :**

```
{
  "DelegateUpdated": {
    "DelegateId": 534,
    "Name": "Manoj",
    "MiddleName": "Subhashbhai",
    "FirstName": "Patel",
    "Title": "Mr.",
    "BadgeNr": "",
    "UserName": "",
    "Password": "",
    "Street": "",
    "StreetNr": "0",
    "PostalCode": "",
    "City": "",
    "Country": "",
    "BirthDate": "",
    "District": "",
    "Email": "",
    "PhoneNr": "",
    "VotingRight": true,
    "VotingWeight": 1,
    "PinId": "9876",
    "FingerPrinterData": "",
    "KeypadLoginCode": "1234"
  }
}
```

### 3.3.7.8 ImportDelegatesFromFile

API Module	Cocon for Plixus Core	Plixus Core
ImportDelegatesFromFile	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Import delegate from the file. (Note: it should be work wit the same version of the installer).

**Structure:**

<http://localhost:8890/CoCon/Delegate/ImportDelegatesFromFile/?FilePath={filePath}>

**Example:**

<http://localhost:8890/CoCon/Delegate/ImportDelegatesFromFile/?FilePath=C:\Users\MP\Desktop\ERS\3.1-05-15-38-fullname.xml>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8 Interactive

This section describes commands that are only applicable for the Televic uniCOS (PlixUs) hardware.

### 3.3.8.1 SetDelegateScreen

Set the delegate screen on the IDCs as well as screen options.  
Only applicable for uniCOS (PlixUs).

#### Parameters applicable for ALL screens:

Name	Description	Type
IsLocked	Lock state of IDC	Bool
IsGoTo	bool indicating whether the IDC should change to the specified screen	Bool
IsShow	Show or hide the screen	Bool

Note that the specific screens described below might have more additional parameters

#### 3.3.8.1.1 Home

API Module	Cocon for Plixus Core	Plixus Core
Home	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Structure:

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Home/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}>

#### Example:

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Home/?IsLocked=True&IsGoTo=True&IsShow=True>

#### Return:

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

#### 3.3.8.1.2 Discussion

API Module	Cocon for Plixus Core	Plixus Core
Discussion	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Additional parameters

Name	Description	Type
Video	Set video viewing mode 0: disabled; no video shown 1: normal 2: full screen	int

#### Structure

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Discussion/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}&Video={int}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Discussion/?IsLocked=True&IsGoTo=True&IsShow=True&Video=1>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.8.1.3 Agenda**

API Module	Cocon for Plixus Core	Plixus Core
Agenda	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Additional parameters**

Name	Description	Type
Select	Select and show active agenda item	bool

**Structure**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Agenda/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}&Select={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Agenda/?IsLocked=True&IsGoTo=True&IsShow=True&Select=True>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.8.1.4 Delegates**

API Module	Cocon for Plixus Core	Plixus Core
Delegates	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Structure**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Delegates/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Delegates/?IsLocked=True&IsGoTo=True&IsShow=True>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.1.5 Voting

API Module	Cocon for Plixus Core	Plixus Core
Voting	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Additional parameters

Name	Description	Type
Res	Show voting results	bool
Graph	Show voting graph	bool

#### Structure

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Voting/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}&Res={bool}&Graph={bool}>

#### Example:

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Voting/?IsLocked=True&IsGoTo=True&IsShow=True&Res=True&Graph=True>

#### Return:

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.1.6 Documents

API Module	Cocon for Plixus Core	Plixus Core
Documents	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Additional parameters

Name	Description	Type
Doc	Id of the document to select	string
Lck	Lock the document (no selection)	bool
Page	Page number to be scrolled to	int
Docfs	Show the document fullscreen	bool

#### Structure

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Documents/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}&Doc={string}&Lck={bool}&Page={int}&Docfs={bool}>

#### Example:

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Documents/?IsLocked=True&IsGoTo=True&IsShow=True&Doc=Cocon-manual&Lck=True&Page=1&Docfs=True>

#### Return:

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.1.7 Messages

API Module	Cocon for Plixus Core	Plixus Core
Messages	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Structure**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Messages/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Messages/?IsLocked=True&IsGoTo=True&IsShow=True>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.1.8 Services

API Module	Cocon for Plixus Core	Plixus Core
Services	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Structure**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Services/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Services/?IsLocked=True&IsGoTo=True&IsShow=True>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.1.9 Configuration

API Module	Cocon for Plixus Core	Plixus Core
Configuration	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Structure**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Configuration/?IsLocked={bool}&IsGoTo={bool}&IsShow={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetDelegateScreen/Configuration/?IsLocked=True&IsGoTo=True&IsShow=True>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.8.2 RemoveScreenLock

API Module	Cocon for Plixus Core	Plixus Core
------------	-----------------------	-------------

RemoveScreenLock	<input checked="" type="checkbox"/>	<input type="checkbox"/>
------------------	-------------------------------------	--------------------------

Remove the screen lock on the Multimedia Unit.  
Only applicable for uniCOS (PlixUs)

**Structure:**

<http://localhost:8890/CoCon/Screen/RemoveScreenLock>

**Return:**

{RemoveScreenLock : {"success" : "true"}}

### 3.3.8.3 SetCustomButtonOnSeat

API Module	Cocon for Plixus Core	Plixus Core
SetCustomButtonOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configure a custom button on a seat

**Structure**

<http://localhost:8890/CoCon/Interactive/SetCustomButtonOnSeat?SeatId={int}&ButtonId={int}&Text={string}&FgColor={string}&BgColor={string}&Show={bool}&EnableDown={bool}&EnableUp={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetCustomButtonOnSeat?SeatId=5&ButtonId=1&Text=RequestFloor&FgColor=black&BgColor=white&Show=True&EnableDown=True&EnableUp=False>

**Return:**

{"SetCustomButtonOnSeat": {"success": "true"}}

### 3.3.8.4 SetCustomButtonOnRoom

API Module	Cocon for Plixus Core	Plixus Core
SetCustomButtonOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configure a custom button on a seat

**Structure**

<http://localhost:8890/CoCon/Interactive/SetCustomButtonOnRoom?ButtonId={int}&Text={string}&FgColor={string}&BgColor={string}&Show={bool}&EnableDown={bool}&EnableUp={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetCustomButtonOnRoom?ButtonId=1&Text=RequestFloor&FgColor=black&BgColor=white&Show=True&EnableDown=True&EnableUp=False>

**Return:**

{"SetCustomButtonOnRoom": {"success": "true"}}

### 3.3.8.5 SetCustomLabelOnSeat

API Module	Cocon for Plixus Core	Plixus Core



SetCustomLabelOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
----------------------	-------------------------------------	--------------------------

Configure a custom button on a seat

**Structure**

<http://localhost:8890/CoCon/Interactive/SetCustomLabelOnSeat?SeatId={int}&LabelId={int}&Text={string}&FgColor={string}&BgColor={string}&Show={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetCustomLabelOnSeat?SeatId=5&LabelId=1&Text=FloorRequested&FgColor=black&BgColor=white&Show=True>

**Return:**

{“SetCustomLabelOnSeat”: {“success”: “true”}}

### 3.3.8.6 SetCustomLabelOnRoom

API Module	Cocon for Plixus Core	Plixus Core
SetCustomLabelOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configure a custom button on a seat

**Structure**

<http://localhost:8890/CoCon/Interactive/SetCustomLabelOnRoom?LabelId={int}&Text={string}&FgColor={string}&BgColor={string}&Show={bool}>

**Example:**

<http://localhost:8890/CoCon/Interactive/SetCustomLabelOnRoom?LabelId=1&Text=FloorRequested&FgColor=black&BgColor=white&Show=True>

**Return:**

{“SetCustomLabelOnRoom”: {“success”: “true”}}

### 3.3.8.7 SetCustomScreenOnSeat

API Module	Cocon for Plixus Core	Plixus Core
SetCustomScreenOnSeat	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configure a custom screen on a seat

**Structure**

<http://localhost:8890/CoCon/Interactive/SetCustomScreenOnSeat?SeatId={int}&Lock={bool}&Go={bool}&Show={bool}&CustomScreenId={int}&Title={string}&Content={String}>

**Example:**

<http://192.168.8.120:8890/CoCon/Interactive/SetCustomScreenOnSeat?SeatId=2&Lock=False&Go=False&Show=True&CustomScreenId=1&Title=CustomTab&Content=<HTML><HEAD><TITLE>UNIT1</TITLE></HEAD><BODY><H1>Custom screen</H1><P>This is a custom screen.</P></BODY></HTML>>

Please note that forward slashes in the “Content” field need to be escaped!

**Return:**

{“SetCustomScreenOnSeat”: {“success”: “true”}}

### 3.3.8.8 SetCustomScreenOnRoom

API Module	Cocon for Plixus Core	Plixus Core
SetCustomScreenOnRoom	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configure a custom button on a seat

#### Structure

<http://localhost:8890/CoCon/Interactive/SetCustomScreenOnRoom?Lock={bool}&Go={bool}&Show={bool}&CustomScreenId={int}&Title={string}&Content={String}>

#### Example:

<http://localhost:8890/CoCon/Interactive/SetCustomScreenOnRoom?Lock=False&Go=False&Show=True&CustomScreenId=1&Title=CustomTab&Content=<HTML><HEAD><TITLE>UNIT1</TITLE></HEAD><BODY><H1>Custom screen</H1><P>This is a custom screen.</P></BODY></HTML>>

Please note that forward slashes in the "Content" field need to be escaped!

#### Return:

```
{"SetCustomScreenOnRoom": {"success": "true"}}
```

### 3.3.9 Audio

#### 3.3.9.1 GetCurrentAudioSoftwareConfiguration

API Module	Cocon for Plixus Core	Plixus Core
GetCurrentAudioSoftwareConfiguration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

#### Structure:

<http://localhost:8890/CoCon/Audio/GetCurrentAudioSoftwareConfiguration>

#### Example:

<http://localhost:8890/CoCon/Audio/GetCurrentAudioSoftwareConfiguration>

#### Return:

```
{"GetCurrentAudioSoftwareConfiguration":{"AudioSoftwareConfiguration":{"Id":3,"Name":"External equalizer"}}
```

#### 3.3.9.2 GetCurrentAudioSoftwareConfigurationInDetails

API Module	Cocon for Plixus Core	Plixus Core
GetCurrentAudioSoftwareConfigurationInDetails	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

#### Structure:

<http://localhost:8890/CoCon/Audio/GetCurrentAudioSoftwareConfigurationInDetails>

#### Example:

<http://localhost:8890/CoCon/Audio/GetCurrentAudioSoftwareConfigurationInDetails>

**Return:**

```
{"GetCurrentAudioSoftwareConfigurationInDetails":{"AudioSoftwareConfiguration":{"Id":1,"Name":"Normal routing","MatrixNodes":[{"InputGroupName":"Floor IN","OutputGroupName":"Floor OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"AUX IN","OutputGroupName":"Floor OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Dante IN","OutputGroupName":"Floor OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Floor IN","OutputGroupName":"AUX OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"AUX IN","OutputGroupName":"AUX OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Dante IN","OutputGroupName":"AUX OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Floor IN","OutputGroupName":"Dante OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"AUX IN","OutputGroupName":"Dante OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Dante IN","OutputGroupName":"Dante OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Lang1 IN","OutputGroupName":"Lang1 OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"Floor IN","OutputGroupName":"Lang1 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"AUX IN","OutputGroupName":"Lang1 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Dante IN","OutputGroupName":"Lang1 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Lang2 IN","OutputGroupName":"Lang1 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Lang3 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang4 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang5 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang6 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang7 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang8 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang9 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang10 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang11 IN","OutputGroupName":"Lang1 OUT","MixValue":33,"Status":false,"BackupStatus":false}, {"InputGroupName":"Lang2 IN","OutputGroupName":"Lang2 OUT","MixValue":0,"Status":true,"BackupStatus":false}, {"InputGroupName":"AUX IN","OutputGroupName":"Lang2 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Dante IN","OutputGroupName":"Lang2 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Lang1 IN","OutputGroupName":"Lang2 OUT","MixValue":0,"Status":false,"BackupStatus":true}, {"InputGroupName":"Lang3 IN","OutputGroupName":"Lang2 OUT","MixValue":33,"Status":false,"BackupStatus":false}]}}}
```









```

OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang11
IN", "OutputGroupName":"Lang11
OUT", "MixValue":0, "Status":true, "BackupStatus":false}, {"InputGroupName":"Floor
IN", "OutputGroupName":"Lang11
OUT", "MixValue":0, "Status":false, "BackupStatus":true}, {"InputGroupName":"AUX
IN", "OutputGroupName":"Lang11
OUT", "MixValue":0, "Status":false, "BackupStatus":true}, {"InputGroupName":"Dante
IN", "OutputGroupName":"Lang11
OUT", "MixValue":0, "Status":false, "BackupStatus":true}, {"InputGroupName":"Lang1
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang2
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang3
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang4
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang5
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang6
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang7
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang8
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang9
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}, {"InputGroupName":"Lang10
IN", "OutputGroupName":"Lang11
OUT", "MixValue":33, "Status":false, "BackupStatus":false}], "AudioGroups": [{"Id":0, "Name":"Floor
IN", "Type":"Input"}, {"Id":1, "Name":"AUX IN", "Type":"Input"}, {"Id":2, "Name":"Dante
IN", "Type":"Input"}, {"Id":3, "Name":"Lang1 IN", "Type":"Input"}, {"Id":4, "Name":"Lang2
IN", "Type":"Input"}, {"Id":5, "Name":"Lang3 IN", "Type":"Input"}, {"Id":6, "Name":"Lang4
IN", "Type":"Input"}, {"Id":7, "Name":"Lang5 IN", "Type":"Input"}, {"Id":8, "Name":"Lang6
IN", "Type":"Input"}, {"Id":9, "Name":"Lang7 IN", "Type":"Input"}, {"Id":10, "Name":"Lang8
IN", "Type":"Input"}, {"Id":11, "Name":"Lang9 IN", "Type":"Input"}, {"Id":12, "Name":"Lang10
IN", "Type":"Input"}, {"Id":13, "Name":"Lang11 IN", "Type":"Input"}, {"Id":1, "Name":"Floor
OUT", "Type":"Output"}, {"Id":2, "Name":"AUX OUT", "Type":"Output"}, {"Id":3, "Name":"Dante
OUT", "Type":"Output"}, {"Id":4, "Name":"Lang1 OUT", "Type":"Output"}, {"Id":5, "Name":"Lang2
OUT", "Type":"Output"}, {"Id":6, "Name":"Lang3 OUT", "Type":"Output"}, {"Id":7, "Name":"Lang4
OUT", "Type":"Output"}, {"Id":8, "Name":"Lang5 OUT", "Type":"Output"}, {"Id":9, "Name":"Lang6
OUT", "Type":"Output"}, {"Id":10, "Name":"Lang7 OUT", "Type":"Output"}, {"Id":11, "Name":"Lang8
OUT", "Type":"Output"}, {"Id":12, "Name":"Lang9 OUT", "Type":"Output"}, {"Id":13, "Name":"Lang10
OUT", "Type":"Output"}, {"Id":14, "Name":"Lang11 OUT", "Type":"Output"}]}]}

```

### 3.3.9.3 GetAudioSoftwareConfigurations

API Module	Cocon for Plixus Core	Plixus Core
GetAudioSoftwareConfigurations	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

#### Structure:

<http://localhost:8890/CoCon/Audio/GetAudioSoftwareConfigurations>

#### Example:

<http://localhost:8890/CoCon/Audio/GetAudioSoftwareConfigurations>

#### Return:

```

{"GetAudioSoftwareConfigurations":{"AudioSoftwareConfigurations":[{"Id":1, "Name":"Normal
routing", "Description":"Incoming audio from microphone ,auxiliary inputs 1 - 2 & 3 and dante input channel 1 is

```



routed by default to the unit speakers ,auxiliary outputs 1 - 2 & 3 and dante output channel 1.", "Predefined":true, "State":"New"}, {"Id":2, "Name":"Distance conferencing", "Description":"Incoming audio from microphone ,auxiliary inputs 1 - 2 & 3 and dante input channel 1 is routed by default to the unit speakers. Incoming audio from microphone is routed by default to the auxiliary outputs 1 - 2 &3 and dante output channel 1.", "Predefined":true, "State":"New"}, {"Id":3, "Name":"External equalizer", "Description":"Incoming audio from auxiliary inputs 1 - 2 & 3 and dante input channel 1 is routed by default to the unit speakers. Incoming audio from microphone is routed by default to the auxiliary outputs 1 - 2 & 3 and dante output channel 1.", "Predefined":true, "State":"New"}, {"Id":4, "Name":"New Preset", "Description":"This is a new preset", "Predefined":false, "State":"Running"}]}}

### 3.3.9.4 GetAudioGroupsInCurrentConfiguration

API Module	Cocon for Plixus Core	Plixus Core
GetAudioGroupsInCurrentConfiguration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This command will reply with all groups from running (Active) Preset.

**Structure:**

<http://localhost:8890/CoCon/Audio/GetAudioGroupsInCurrentConfiguration>

**Example:**

<http://localhost:8890/CoCon/Audio/GetAudioGroupsInCurrentConfiguration>

**Return:**

```
{
  "GetAudioGroupsInCurrentConfiguration": {
    "AudioGroups": [
      { "Id": 0, "Name": "Floor IN", "Type": "Input" },
      { "Id": 1, "Name": "AUX IN", "Type": "Input" },
      { "Id": 2, "Name": "Dante IN", "Type": "Input" },
      { "Id": 3, "Name": "Lang1 IN", "Type": "Input" },
      { "Id": 4, "Name": "Lang2 IN", "Type": "Input" },
      { "Id": 5, "Name": "Lang3 IN", "Type": "Input" },
      { "Id": 6, "Name": "Lang4 IN", "Type": "Input" },
      { "Id": 7, "Name": "Lang5 IN", "Type": "Input" },
      { "Id": 8, "Name": "Lang6 IN", "Type": "Input" },
      { "Id": 9, "Name": "Lang7 IN", "Type": "Input" },
      { "Id": 10, "Name": "Lang8 IN", "Type": "Input" },
      { "Id": 11, "Name": "Lang9 IN", "Type": "Input" },
      { "Id": 12, "Name": "Lang10 IN", "Type": "Input" },
      { "Id": 13, "Name": "Lang11 IN", "Type": "Input" },
      { "Id": 1, "Name": "Floor OUT", "Type": "Output" },
      { "Id": 2, "Name": "AUX OUT", "Type": "Output" },
      { "Id": 3, "Name": "Dante OUT", "Type": "Output" },
      { "Id": 4, "Name": "Lang1 OUT", "Type": "Output" },
      { "Id": 5, "Name": "Lang2 OUT", "Type": "Output" },
      { "Id": 6, "Name": "Lang3 OUT", "Type": "Output" },
      { "Id": 7, "Name": "Lang4 OUT", "Type": "Output" },
      { "Id": 8, "Name": "Lang5 OUT", "Type": "Output" },
      { "Id": 9, "Name": "Lang6 OUT", "Type": "Output" },
      { "Id": 10, "Name": "Lang7 OUT", "Type": "Output" },
      { "Id": 11, "Name": "Lang8 OUT", "Type": "Output" },
      { "Id": 12, "Name": "Lang9 OUT", "Type": "Output" },
      { "Id": 13, "Name": "Lang10 OUT", "Type": "Output" },
      { "Id": 14, "Name": "Lang11 OUT", "Type": "Output" }
    ]
  }
}
```

### 3.3.9.5 SetAudioSoftwareConfigurationById

API Module	Cocon for Plixus Core	Plixus Core
SetAudioSoftwareConfigurationById	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Parameters:**

Name	Description	Type
Id	The id of the audio software configuration	int

**Structure:**

<http://localhost:8890/CoCon/Audio/SetAudioSoftwareConfigurationById/?Id={id}>

**Example:**

<http://localhost:8890/CoCon/Audio/SetAudioSoftwareConfigurationById/?Id=1>

**Return:**

```
{"SetAudioSoftwareConfigurationById":{"AudioSoftwareConfiguration":{"Description":"Incoming audio from microphone, auxiliary inputs 1, 2 & 3 and Dante input channel 1 is routed by default to the unit speakers, auxiliary outputs 1, 2 & 3 and Dante output channel 1.", "Id":1, "Name":"Normal routing", "Predefined":true, "State":"Active"}}
```

**3.3.9.6 SetAudioSoftwareConfigurationByName**

API Module	Cocon for Plixus Core	Plixus Core
SetAudioSoftwareConfigurationByName	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Name: SetAudioSoftwareConfigurationByName

**Parameters:**

Name	Description	Type
Name	The Name of the audio software configuration	string

**Structure:**

<http://localhost:8890/CoCon/Audio/SetAudioSoftwareConfigurationByName/?Name={name}>

**Example:**

<http://localhost:8890/CoCon/Audio/SetAudioSoftwareConfigurationByName/?Name=Normal routing 123>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**3.3.9.7 ChangeMatrixNodeStatus**

API Module	Cocon for Plixus Core	Plixus Core
ChangeMatrixNodeStatus	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This command will apply on the current running (active) preset only.

**Parameters:**

Name	Description	Type
InputGroupName	The name of routing input group	string
OutputGroupName	The name of routing output group	string
Status	The matrix node status(enable or not) True/False	bool
BackupStatus	The matrix node backup status(enable or not) True/False	bool
MixValue	The mix value of the group(-100-0)	int

**Structure:**

<http://localhost:8890/CoCon/Audio/ChangeMatrixNodeStatus/?InputGroupName={inputGroupName}&OutputGroupName={outputGroupName}&Status={status}&BackupStatus={backupStatus}&MixValue={mixValue}>

**Example:**

<http://localhost:8890/CoCon/Audio/ChangeMatrixNodeStatus/?InputGroupName=Floor In&OutputGroupName=Floor Out&Status=False&BackupStatus=False&MixValue=-20>

**Return:**

```
{"ChangeMatrixNodeStatus":{"MatrixNode":{"InputGroupName":"Floor In","OutputGroupName":"Floor out","MixValue":-20,"Status":false,"BackupStatus":false,"Id":201}}}
```

**3.3.9.8 GetMatrixNodeStatus**

API Module	Cocon for Plixus Core	Plixus Core
GetMatrixNodeStatus	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get the status of a specific node in running (Active) audio preset. The input group and output group are required.

Name	Description	Type
InputGroupName	The name of routing input group	String
OutputGroupName	The name of routing output group	string

**Structure:**

<http://localhost:8890/CoCon/Audio/GetMatrixNodeStatus/?InputGroupName={inputGroupName}&OutputGroupName={inputGroupName}>

**Example:**

<http://localhost:8890/CoCon/Audio/GetMatrixNodeStatus/?InputGroupName=Floor In&OutputGroupName=Floor Out>

**Return:**

```
{"GetMatrixNodeStatus":{"MatrixNode":{"InputGroupName":"Floor In","OutputGroupName":"Floor out","MixValue":-20,"Status":false,"BackupStatus":true,"Id":201}}}
```

**3.3.9.9 SaveAudioSoftwareConfiguration**

API Module	Cocon for Plixus Core	Plixus Core
SaveAudioSoftwareConfiguration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

If the " name " corresponding configuration not exists in the database, it will create a new configuration and If the " name " is exists in the database then save it.

Note: CoCon will ignore the text case for the preset name.

**Parameters:**

Name	Description	Type
Name	Name of the configuration	string
Description	Description of the configuration	string

**Structure:**

<http://localhost:8890/CoCon/Audio/SaveAudioSoftwareConfiguration/?Name={name}&Description={description}>

**Example:**

<http://localhost:8890/CoCon/Audio/SaveAudioSoftwareConfiguration/?Name=Audio Preset Test&Description=Audio Preset Test>

**Return:**

```
{"SaveAudioSoftwareConfiguration":{"AudioSoftwareConfiguration":{"Id":16,"Name":"Audio Preset Test","Description":"Audio Preset Test","Predefined":false,"State":"New"}}
```

**Return for CoCon for Plixus Core & Plixus Core:**

Note: only the second reply contains the right data, so user have to send 2 time api call to get the updated info.

### 3.3.10 Logging

#### 3.3.10.1 GetCUState

API Module	Cocon for Plixus Core	Plixus Core
GetCUState	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get the status of the Plixus engine

**Structure:**

<http://localhost:8890/CoCon/Logging/GetCUState>

**Return:**

```
{"GetCUState":{"CuState":"Connected"}}  
{"GetCUState":{"CuState":"Disconnected"}}
```

#### 3.3.10.2 GetUnitErrorState

API Module	Cocon for Plixus Core	Plixus Core
GetUnitErrorState	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Gets the IDs of the units which are currently in error state.

**Structure:**

<http://localhost:8890/CoCon/Logging/GetUnitErrorState>

**Return:**

```
{"GetUnitErrorState":{"Units":[{"UnitId":1},{"UnitId":2},{"UnitId":6},{"UnitId":7}]}}
```

### 3.3.11 ButtonLED\_Event

#### 3.3.11.1 SetButtonsEnabled

API Module	Cocon for Plixus Core	Plixus Core
SetButtonsEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the buttons of one or more seats to enabled or disabled.  
 If the buttons are disabled, no button events are possible.  
 Set the enabled to true to allow CoCon, and the API, to receive button events from the units.

The event "ButtonPressed" (see section 3.2.10.1) can only be received by the API clients after one or more of the buttons have been enabled, using the command "SetButtonsEnabled".

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetButtonsEnabled/?SeatNr=<SeatNr>&ButtonNr=<ButtonNr>&Enabled=<Enabled>](http://localhost:8890/CoCon/ButtonLED_Event/SetButtonsEnabled/?SeatNr=<SeatNr>&ButtonNr=<ButtonNr>&Enabled=<Enabled>)

**Parameters:**

Name	Description	Type
SeatNr	The number of the seat on which the buttons are enable/disabled. Use 9999 for all units	int
ButtonNr	The number of the button which was pressed. Possible values: 1-5, depending on the possibilities of the delegate units.  For example: delegate units with 3 buttons only have buttons 2-3-4	int
Enabled	True to enable buttons False to disable buttons	bool

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetButtonsEnabled/?SeatNr=5&ButtonNr=2&Enabled=true](http://localhost:8890/CoCon/ButtonLED_Event/SetButtonsEnabled/?SeatNr=5&ButtonNr=2&Enabled=true)

**Return:**

0 → operation success; 1 → Operation fail (not valid value); -1 → exception

**3.3.11.2 SetLED**

API Module	Cocon for Plixus Core	Plixus Core
SetLED	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Sets the LED of one or more seats.

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetLED/?SeatNr=<SeatNr>&LEDNr=<LEDNr>&State=<State>](http://localhost:8890/CoCon/ButtonLED_Event/SetLED/?SeatNr=<SeatNr>&LEDNr=<LEDNr>&State=<State>)

**Parameters:**

Name	Description	Type
SeatNr	The number of the seat on which the LEDs are set. Use 9999 for all units	int
LEDNr	The number of the LED which should be set. Possible values: 1-5, depending on the possibilities of the delegate units.	int

	For example: delegate units with 3 buttons only have buttons 2-3-4	
State	The state to which the LED should be set Values: On Off Blinking	string

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetLED/?SeatNr=4&LEDNr=2&State=Blinking](http://localhost:8890/CoCon/ButtonLED_Event/SetLED/?SeatNr=4&LEDNr=2&State=Blinking)

**Return:**

0 → operation success; 1 → Operation fail (not valid value); -1 → exception

### 3.3.11.3 SetNextButtonEnabled

API Module	Cocon for Plixus Core	Plixus Core
SetNextButtonEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Enables the Next button of a unit. When enabled and a button up event is triggered, an API event will be sent out. The events will remain active for each button up event until the next system reboot or until they are explicitly de-activated through this API call.

Note: This api call will work only on delegate(no chairmen) seat where unit have Next & Prior physical buttons and configurable by Room server xml file.

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetNextButtonEnabled/?SeatNr={SeatNr}&Enabled={Enabled}](http://localhost:8890/CoCon/ButtonLED_Event/SetNextButtonEnabled/?SeatNr={SeatNr}&Enabled={Enabled})

**Parameters:**

Name	Description	Type
SeatNr	The seat number	int
Enabled	true, false	boolean

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetNextButtonEnabled/?SeatNr=7&Enabled=true](http://localhost:8890/CoCon/ButtonLED_Event/SetNextButtonEnabled/?SeatNr=7&Enabled=true)

**Return:**

0 → operation success; 1 → Operation fail (not valid value); -1 → exception

### 3.3.11.4 SetPriorButtonEnabled

API Module	Cocon for Plixus Core	Plixus Core
SetPriorButtonEnabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Enables the Prior button of a unit. When enabled and a button up event is triggered, an API event will be sent out. The events will remain active for each button up event until the next system reboot or until they are explicitly de-activated through this API call.

Note: This api call will work only on delegate(no chairmen) seat where unit have Next & Prior physical buttons and configurable by Room server xml file..

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetPriorButtonEnabled/?SeatNr={SeatNr}&Enabled={Enabled}](http://localhost:8890/CoCon/ButtonLED_Event/SetPriorButtonEnabled/?SeatNr={SeatNr}&Enabled={Enabled})

**Parameters:**

Name	Description	Type
SeatNr	The seat number	int
Enabled	true, false	boolean

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetPriorButtonEnabled/?SeatNr=7&Enabled=true](http://localhost:8890/CoCon/ButtonLED_Event/SetPriorButtonEnabled/?SeatNr=7&Enabled=true)

**Return:**

0 → operation success; 1 → Operation fail (not valid value); -1 → exception

**3.3.11.5 SetNextButtonLedStatus**

API Module	Cocon for Plixus Core	Plixus Core
SetNextButtonLedStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Sets the Next button LED status

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetNextButtonLedStatus/?UnitId={unitId}&Status={status}](http://localhost:8890/CoCon/ButtonLED_Event/SetNextButtonLedStatus/?UnitId={unitId}&Status={status})

**Parameters:**

Name	Description	Type
UnitId	The unit's Id	int
Status	we provide five status : On, Off, Fast, Slow, Medium	string

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetNextButtonLedStatus/?UnitId=1&Status=on](http://localhost:8890/CoCon/ButtonLED_Event/SetNextButtonLedStatus/?UnitId=1&Status=on)

**Return:**

0 → operation success; 1 → Operation fail (not valid value); -1 → exception

**3.3.11.6 SetPriorButtonLedStatus**

API Module	Cocon for Plixus Core	Plixus Core
SetPriorButtonLedStatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Sets the Prior button LED status

**Structure:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetPriorButtonLedStatus/?UnitId={unitId}&Status={status}](http://localhost:8890/CoCon/ButtonLED_Event/SetPriorButtonLedStatus/?UnitId={unitId}&Status={status})

**Parameters:**

Name	Description	Type
------	-------------	------

UnitId	The unit's Id	int
Status	we provide five status : On, Off, Fast, Slow, Medium	string

**Example:**

[http://localhost:8890/CoCon/ButtonLED\\_Event/SetPriorButtonLedStatus/?UnitId=7&Status=fast](http://localhost:8890/CoCon/ButtonLED_Event/SetPriorButtonLedStatus/?UnitId=7&Status=fast)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.12 External

#### 3.3.12.1 Send message to Operator

API Module	Cocon for Plixus Core	Plixus Core
SendMsgToOperato	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Structure:**

<http://localhost:8890/CoCon/External/SendMsgToOperator/?Message={msg}>

**Example:**

[http://localhost:8890/CoCon/External/SendMsgToOperator/?Message={"This is an example msg"}](http://localhost:8890/CoCon/External/SendMsgToOperator/?Message={)

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

#### 3.3.12.2 NotifyExternalCall

API Module	Cocon for Plixus Core	Plixus Core
NotifyExternalCall	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Notification to Interpreter Desk for the external call.

**Structure:**

<http://localhost:8890/CoCon/External/NotifyExternalCall/?To={unitIds}&Beep={isBeep}>

**Parameters:**

Name	Description	Type
UnitIds	The unit ids list separate by comma. 2,5,10	srting
Beep	True/ False To send Beep sound on the desk	bool

**Example:**

<http://localhost:8890/CoCon/External/NotifyExternalCall/?To=2,3,4&Beep=True>

**Returns:** string

{"NotifyExternalCall":true}



### 3.3.13 Interpretation

Interpreter management and configuration commands

#### 3.3.13.1 GetAllBoothsInRoom

API Module	Cocon for Plixus Core	Plixus Core
GetAllBoothsInRoom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all booths name from current active room.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetAllBoothsInRoom>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetAllBoothsInRoom>

**Return:**

```
{"GetAllBoothsInRoom":{"Booths":[{"BoothId":1,"Desks":[10,7]]}}
```

#### 3.3.13.2 AddLanguage

API Module	Cocon for Plixus Core	Plixus Core
AddLanguage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add new language to DB.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/AddLanguage/?Name={name}&ISO={iso}>

**Parameters:**

Name	Description	Type
Name	Name of the Language	string
ISO	2/3 letters code for the Language	string

**Return:** int

**Example:**

<http://localhost:8890/CoCon/Interpretation/AddLanguage/?Name=English&ISO=EN>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

#### 3.3.13.3 RemoveLanguage

API Module	Cocon for Plixus Core	Plixus Core
RemoveLanguage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Remove language from the existing list of language.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/RemoveLanguage/?Name={name}>

**Parameters:**

Name	Description	Type
Name	Name of the Language	string

**Return:** int

**Example:**

<http://localhost:8890/CoCon/Interpretation/RemoveLanguage/?Name=English>

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.4 ClearLanguages

API Module	Cocon for Plixus Core	Plixus Core
ClearLanguages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Clear all the language from the language list.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/ClearLanguages>

**Return:** int

**Example:**

<http://localhost:8890/CoCon/Interpretation/ClearLanguages>

**Return:**

0 → operation success (reply not send in Cocon for Plixus Core and Plixus Core); 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.5 AddInterpreterPreset

API Module	Cocon for Plixus Core	Plixus Core
AddInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add new interpreter preset

**Structure:**

<http://localhost:8890/CoCon/Interpretation/AddInterpreterPreset/?Name={name}&Description={description}&MicMode={micMode}&BoothMode={boothMode}>

**New Optional Structure :**

<http://localhost:8890/CoCon/Interpretation/AddInterpreterPreset/?Name={name}&Description={description}&MicMode={micMode}&BoothMode={boothMode}&MultipleChannelActivation={multipleChannelActivation}&IsOverrideChannel={isOverrideChannel}&IsAllActiveMicsLEDBlink={isAllActiveMicsLEDBlink}&AllActiveMicsLEDBlinkTime={allActiveMicsLEDBlinkTime}&IsSlowDownImplementedExternally={isSlowDownImplementedExternally}&IsListenerCountConfigurator={isListenerCountConfigurator}&ListenerCountConfiguratorTime={listenerCountConfiguratorTime}>

**Parameters:**

Name	Description	Type
Name	The name of Interpreter preset (Required)	string
Description	The description of the interpreter preset (Optional)	string

MicMode	Interpretation mic mode 0 = override mode 1 = mixed mode (Optional)	int
BoothMode	Interpretation mode between booths 0 = override mode 1 = mixed mode 2 = Lock on A. Override on B (Optional)	Int
MultipleChannelActivation	Allow multiple channel can active on same booth (True / False : Optional parameter, Default value : False)	Bool
IsOverrideChannel	Allow only one channel can active in one booth (True / False : Optional parameter, Default value : False)	Bool
IsAllActiveMicsLEDBlink	All active mics LED blinking if the speak slow function enabled (True / False : Optional parameter, Default value : False)	Bool
AllActiveMicsLEDBlinkTime	Speak slow function Enabled + configure the time for the mic led blinking for speak slow function. (Optional parameter, Default value : 5 sec)	int
IsSlowDownImplementedExternally	Send notification for the speak low to external API user. (True / False : Optional parameter, Default value : True)	Bool
IsListenerCountConfigurator	Listener Count function enabled (True / False : Optional parameter, Default value : False)	Bool
ListenerCountConfiguratorTime	Configure the time for the listener count event from CU to Server (Optional parameter, Default value : 60 sec)	int

**Example:**

<http://localhost:8890/CoCon/Interpretation/AddInterpreterPreset/?Name=My first test &Description=Test first test&MicMode=0&BoothMode=0>

**Return:**

```
{"AddInterpreterPreset":{"InterpreterPreset":{"Id":11,"Name":"My first test ","Description":"Test first test","DeskMode":0,"BoothMode":0,"State":"Inactive"}}
```

**New Return value :**

```
{"AddInterpreterPreset":{"InterpreterPreset":{"Id":10,"Name":" MTest 1","Description":"","DeskMode":0,"BoothMode":0,"MultipleChannelActivation":false,"IsOverrideChannel":false,"IsAllActiveMicsLEDBlink":false,"AllActiveMicsLEDBlinkTime":5,"IsSlowDownImplementedExternally":true,"IsListenerCountConfigurator":false,"ListenerCountConfiguratorTime":60,"State":"Inactive"}}
```

### 3.3.13.6 DeleteInterpreterPreset

API Module	Cocon for Plixus Core	Plixus Core
DeleteInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Delete the existing interpreter preset by Preset id

**Structure:**

<http://localhost:8890/CoCon/Interpretation/DeleteInterpreterPreset/?Id={presetId}>

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int

**Return:** int

Note: Active preset can't be deleted.

**Example:**

<http://localhost:8890/CoCon/Interpretation/DeleteInterpreterPreset/?Id=2>

**Return:**

0 → operation success (reply not send in Cocon for Plixus Core and Plixus Core); 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.7 ApplyInterpreterPreset

API Module	Cocon for Plixus Core	Plixus Core
ApplyInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Send interpreter Preset to activate preset by ID

**Structure:**

<http://localhost:8890/CoCon/Interpretation/ApplyInterpreterPreset/?Id={presetId}>

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int

**Return:** int

**Example:**

<http://localhost:8890/CoCon/Interpretation/ApplyInterpreterPreset/?Id=2>

**Return:**

0 → operation success (reply not send in Plixus Core); 1 → Operation fail (not valid value) -1 → exception

**Return for CoCon for Plixus Core & Plixus Core:**

Note: Api call will work as expected and update event will recived instead of "0"

String : {"ApplyInterpreterPreset":2}

### 3.3.13.8 GetActiveInterpreterPreset

API Module	Cocon for Plixus Core	Plixus Core
GetActiveInterpreterPreset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get current active interpreter preset information.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetActiveInterpreterPreset>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetActiveInterpreterPreset>

**Return:** string

```

{"GetActiveInterpreterPreset":{"InterpreterPreset":{"Id":0,"Name":"Default","Description":"The default configuration with English, French, German and Dutch.", "DeskMode":0,"BoothMode":0,"MultiChannelActivation":false,"Locked":false,"AudienceIndication":false,"AudienceIndicationInterval":60,"State":"Active","Channels":[{"Order":1,"Language":"01 EN","Booth":"1","Desks":[{"Name":"1","ActiveChannel":0,"Relays":["Floor","Floor","Floor","Floor","Floor","Floor"]}], "Quality":0,"IsRecording":false}, {"Order":2,"Language":"02 FR","Booth":"EmptyBooth","Desks":[],"Quality":0,"IsRecording":false}, {"Order":3,"Language":"03 DE","Booth":"EmptyBooth","Desks":[],"Quality":0,"IsRecording":false}, {"Order":4,"Language":"04 NL","Booth":"EmptyBooth","Desks":[],"Quality":0,"IsRecording":false}, {"Order":5,"Language":"05 TA","Booth":"EmptyBooth","Desks":[],"Quality":0,"IsRecording":false}]}

```

### 3.3.13.9 GetAllInterpreterPresets

API Module	Cocon for Plixus Core	Plixus Core
GetAllInterpreterPresets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all interpreter presets from the current room.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetAllInterpreterPresets>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetAllInterpreterPresets>

**Return:** string

```

{"GetAllInterpreterPresets":{"InterpreterPreset":{"Id":7,"Name":"Test 1","Description":"","DeskMode":0,"BoothMode":0,"MultipleChannelActivation":false,"IsOverrideChannel":true,"IsAllActiveMicsLEDBlink":true,"AllActiveMicsLEDBlinkTime":5,"IsSlowDownImplementedExternally":true,"IsListenerCountConfigurator":true,"ListenerCountConfiguratorTime":10,"State":"Active"}, {"Id":9,"Name":"My first test","Description":"Test first test","DeskMode":0,"BoothMode":0,"MultipleChannelActivation":false,"IsOverrideChannel":false,"IsAllActiveMicsLEDBlink":false,"AllActiveMicsLEDBlinkTime":5,"IsSlowDownImplementedExternally":true,"IsListenerCountConfigurator":false,"ListenerCountConfiguratorTime":60,"State":"Inactive"}

```

### 3.3.13.10 GetAllLanguages

API Module	Cocon for Plixus Core	Plixus Core
GetAllLanguages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all languages which is available in language list.

**Structure :**

<http://localhost:8890/CoCon/Interpretation/GetAllLanguages>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetAllLanguages>

**Return:** string

```

{"GetAllLanguages":{"Languages":[{"Name":"Bulgarian","ISO":"BG"}, {"Name":"Bosnian","ISO":"BS"}, {"Name":"Bihari","ISO":"BH"}, {"Name":"Czech","ISO":"CS"}, {"Name":"Cree","ISO":"CR"}, {"Name":"Dutch","ISO":"NL"}, {"Name":"Divehi","ISO":"DV"}, {"Name":"Hebrew","ISO":"HE"}, {"Name":"Haitian","ISO":"HT"}, {"Name":"Limburgish","ISO":"LI"}]}

```

### 3.3.13.11 GetAllChannels

API Module	Cocon for Plixus Core	Plixus Core
GetAllChannels	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get all channels list by existing preset id.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetAllChannels/?PresetId={presetId}>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetAllChannels/?PresetId=2>

**Return:** String

```
{
  "GetAllChannels": {
    "Channels": [
      {
        "Order": 1,
        "Language": "English",
        "AudioRouting": "1",
        "DisplayText": "01 EN",
        "AutoFloor": false,
        "BackupChannel": "Floor"
      },
      {
        "Order": 2,
        "Language": "Chinese",
        "AudioRouting": "2",
        "DisplayText": "02 ZH",
        "AutoFloor": false,
        "BackupChannel": "Floor"
      },
      {
        "Order": 3,
        "Language": "Dutch",
        "AudioRouting": "3",
        "DisplayText": "03 NL",
        "AutoFloor": false,
        "BackupChannel": "Floor"
      },
      {
        "Order": 4,
        "Language": "French",
        "AudioRouting": "External",
        "DisplayText": "04 FR",
        "AutoFloor": false,
        "BackupChannel": "Floor"
      },
      {
        "Order": 5,
        "Language": "Hindi",
        "AudioRouting": "Disabled",
        "DisplayText": "05 HI",
        "AutoFloor": false,
        "BackupChannel": "Floor"
      }
    ]
  }
}
```

### 3.3.13.12 AddChannel

API Module	Cocon for Plixus Core	Plixus Core
AddChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add channel to existing preset via preset Id.

Please note that for now, channels should always be added at the end of the channel list (i.e., you have to supply the next free channel number to the “Order” parameter.

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int
Order	Number of the channels	int
Language	Language name	string
DisplayText	Display text: Text that is shown on the channel selector. E.g.: 02 ENG	string
AudioRouting	Set audio routing for the channel. Plixus Core & CoCon For Plixus Core: 'comma-separated-booth-numbers', Disabled, External, EmptyBooth	string
BackupChannel	Set the backup channel Floor (Now only floor is backup channel) Optional parameter.	string
AutoFloor	Set auto floor is enabled or disabled False (CoCon 5.1 does not support auto floor yet) Optional parameter.	bool
IsRecording	Channel recording or not. Optional parameter.	bool

**Structure:**

<http://localhost:8890/CoCon/Interpretation/AddChannel/?PresetId={presetId}&Order={channelNumber}&Language={language}&DisplayText={displayText}&AudioRouting={audioRouting}&BackupChannel={backupChannel}&AutoFloor={autoFloor}&IsRecording={IsRecording}>

**Example:**

**CoCon for Core & Plixus Core:**

<http://192.168.8.120:8890/CoCon/Interpretation/AddChannel/?PresetId=5&Order=4&Language=English&DisplayText=EN&AudioRouting=1,2&BackupChannel=Floor&AutoFloor=False&IsRecording=False>

**Return: string**

**Note 1:** The language has to be added in the language list otherwise the API will return: “>>API command not executed correctly, Please check the Language is existing or not”

**Note 2:** When you add a channel with a non-consecutive order number CoCon will change the order number automatically to the next available order. This will be stated in the return:

**Return:**

**CoCon for Core & Plixus Core:**

```
"AddChannel":{"Channel":{"Order":1,"Language":"English","AudioRouting":"1,2","DisplayText":"EN","AutoFloor":false,"IsRecording":false,"BackupChannel":"Floor"}}
```

\*\*Log file will have detail for wrong order : You gave an inappropriate channel order, and it has been changed from 5 to 3!

**3.3.13.13 EditChannel**

API Module	Cocon for Plixus Core	Plixus Core
EditChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Edit the existing channel via preset id. (Overwrite the existing channel.)

**Please note that it is not allowed to change the Language Name for a channel.**

**Structure:**

<http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId={presetId}&Order={channelNumber}&Language={language}&DisplayText={displayText}&AudioRouting={audioRouting}&BackupChannel={backupChannel}&AutoFloor={autoFloor}&IsRecording={IsRecording}>

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int
Order	Order of the channel	int
Language	Language name	string
DisplayText	Display text: Text that is shown on the channel selector. E.g.: 02 ENG	string
AudioRouting	Set audio routing for the channel. Plixus Core & CoCon For Plixus Core: 'comma-separated-booth-numbers', Disabled, External, EmptyBooth	string
BackupChannel	Set the backup channel Floor (Now only floor is backup channel)	string
AutoFloor	Set auto floor is enabled or disabled False (CoCon 4.0 does not support auto floor yet)	bool

IsRecording	Channel recording or not. Optional parameter.	bool
-------------	--	------

**Return: string**

```
{"EditChannel":{"Channel":{"Order":4,"Language":"Hindi","AudioRouting":"External","DisplayText":"04 HI","AutoFloor":false,"BackupChannel":"Floor","IsRecording":false}}}
```

**Example:**

<http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=4&Language=Hindi&DisplayText=HI&AudioRouting=External&BackupChannel=Floor&AutoFloor=False&IsRecording=False>

This command can be used to swap languages between 2 booths. For example, we start from a situation where Booth 1 is English and Booth 2 is French. If we want to swap them, we need to execute the following sequence of commands:

- Change the audio routing for the English channel to “Disabled” (“freeing” Booth 1):  
[http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=1&Language=English&DisplayText=01\\_EN&AudioRouting=Disabled&BackupChannel=Floor&AutoFloor=False](http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=1&Language=English&DisplayText=01_EN&AudioRouting=Disabled&BackupChannel=Floor&AutoFloor=False)
- Move the French channel to Booth 1:  
[http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=2&Language=French&DisplayText=01\\_FR&AudioRouting=1&BackupChannel=Floor&AutoFloor=False](http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=2&Language=French&DisplayText=01_FR&AudioRouting=1&BackupChannel=Floor&AutoFloor=False)
- Move the English channel to Booth 2:  
[http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=1&Language=English&DisplayText=02\\_EN&AudioRouting=2&BackupChannel=Floor&AutoFloor=False](http://localhost:8890/CoCon/Interpretation/EditChannel/?PresetId=21&Order=1&Language=English&DisplayText=02_EN&AudioRouting=2&BackupChannel=Floor&AutoFloor=False)

### 3.3.13.14 DeleteChannel

API Module	Cocon for Plixus Core	Plixus Core
DeleteChannel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Delete channel form the preset via channel id.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/DeleteChannel/?PresetId={presetId}&Order={ordernumber}>

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int
Order	The order of the channel	int

**Example:**

<http://localhost:8890/CoCon/Interpretation/DeleteChannel/?PresetId=21&Order=18>

**Return: int**

0 → operation success (reply not send in Cocon for Plixus Core and Plixus Core); 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.15 RemoveAllChannels

API Module	Cocon for Plixus Core	Plixus Core
RemoveAllChannels	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



Remove all the channels from the existing preset.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/RemoveAllChannels/?PresetId={presetId}>

**Parameters:**

Name	Description	Type
PresetId	The Id of Interpreter Preset	int

**Example:**

<http://localhost:8890/CoCon/Interpretation/RemoveAllChannels/?PresetId=21>

**Return:** int

0 → operation success (reply not send in Cocon for Plixus Core and Plixus Core); 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.16 SetDisplayText

API Module	Cocon for Plixus Core	Plixus Core
SetDisplayTex	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set / modify display text for the existing channel via channel Id.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/SetDisplayText/?PresetId={presetId}Order={order}&NewText={text}>

**Parameters:**

Name	Description	Type
PresetId	The preset of the Id which will be used	int
Order	The order of the channel	int
NewText	Text for display on channel selector.	string

**Example:**

<http://localhost:8890/CoCon/Interpretation/SetDisplayText/?PresetId=12&Order=2&NewText=ENG>

**Return:** int

0 → operation success (reply not send in Cocon for Plixus Core and Plixus Core); 1 → Operation fail (not valid value) -1 → exception

### 3.3.13.17 SaveInterpreterPresetAs

API Module	Cocon for Plixus Core	Plixus Core
SaveInterpreterPresetAs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

It will save the preset with new name, based on the existing preset id.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/SaveInterpreterPresetAs/?Id={id}&Name={name}&Description={description}&MicMode={micMode}&BoothMode={boothMode}>

**New Structure with additional optional fields :**

<http://localhost:8890/CoCon/Interpretation/SaveInterpreterPresetAs/?Id={id}&Name={name}&Description={description}&MicMode={micMode}&BoothMode={boothMode}&MultipleChannelActivation={multipleChannelActivation}&IsOverrideChannel={isOverrideChannel}&IsAllActiveMicsLEDBlink={isAllActiveMicsLEDBlink}&AllActiveMicsLEDBlinkTime={allActiveMicsLEDBlinkTime}&IsSlowDownImplementedExternally={isSlowDownImplementedExternally}&IsListenerCountConfigurator={isListenerCountConfigurator}&ListenerCountConfiguratorTime={listenerCountConfiguratorTime}>

**Parameters:**

Name	Description	Type
Id	Existing Preset's id (Required)	int
Name	The name of Interpreter preset (Required)	string
Description	The description of the interpreter preset (Optional)	string
MicMode	Interpretation mic mode 0 = override mode 1 = mixed mode	int
BoothMode	Interpretation mode between booths 0 = override mode 1 = mixed mode 2 = Lock on A. Override on B (Optional)	Int
MultipleChannelActivation	Allow multiple channel can active on same booth (True / False : Optional parameter, Default value : False) (Optional)	Bool
IsOverrideChannel	Allow only one channel can active in one booth (True / False : Optional parameter, Default value : False)	Bool
IsAllActiveMicsLEDBlink	All active mics LED blinking if the speak slow function enabled (True / False : Optional parameter, Default value : False)	Bool
AllActiveMicsLEDBlinkTime	Speak slow function Enabled + configure the time for the mic led blinking for speak slow function. (Optional parameter, Default value : 5 sec)	int
IsSlowDownImplementedExternally	Send notification for the speak low to external API user. (True / False : Optional parameter, Default value : True)	Bool
IsListenerCountConfigurator	Listener Count function enabled (True / False : Optional parameter, Default value : False)	Bool
ListenerCountConfiguratorTime	Configure the time for the listener count event from CU to Server (Optional parameter, Default value : 60 sec)	int

**Return:**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

```
{
  "ActivePresetUpdated": {
    "InterpreterPreset": {
      "InterpreterPreset": {
        "Id": 4,
        "Name": "New Preset 1",
        "Description": "",
        "DeskMode": 1,
        "BoothMode": 0,
        "MultipleChannelActivation": true,
        "IsOverrideChannel": false,
        "IsAllActiveMicsLEDBlink": true,
        "AllActiveMicsLEDBlinkTime": 5,
        "IsSlowDownImplementedExternally": true,
        "IsListenerCountConfigurator": true,
        "ListenerCountConfiguratorTime": 60,
        "State": "Active",
        "Channels": [
          {
            "Order": 1,
            "Language": "English",
            "Booth": "1",
            "Desks": [
              {
                "Name": "1",
                "ActiveChannel": 0,
                "Relays": [],
                "Quality": 3,
                "IsRecording": false
              }
            ]
          },
          {
            "Order": 2,
            "Language": "Chinese",
            "Booth": "EmptyBooth",
            "Desks": [],
            "Quality": 3,
            "IsRecording": false
          }
        ]
      }
    }
  }
}
```

**Note: CoCon for Plexus Core & PlexusCore:**

This call will update the preset based on the Preset id, not create new preset. (Function like update existing preset).

### 3.3.13.18 GetAllChannelRealtimeData

API Module	Cocon for Plixus Core	Plixus Core
GetAllChannelRealtimeData	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get real time data for channels based on the active interpreter preset.

Note: user should request for this command.

The quality of the channel

	Cocon for Core / Plixus Core
floor	0
translation from floor	1
2nd translation from floor	2
3rd translation from floor	3
4th translation from floor	4
...	
Listening to own translation	255

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetAllChannelRealtimeData>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetAllChannelRealtimeData>

**Return:**

**CoCon for Core & Plixus Core:**

```
{"GetAllChannelRealTimeData":[{"Order":1,"Language":"Hindi","Booths":[]},{"Order":2,"Language":"English","Booths":[{"Name":1,"Desks":[{"Name":1,"ActiveChannel":1,"Relay":"English"},"Quality":255}]}]}
```

### 3.3.13.19 GetSpeakSlowerRequestList

API Module	Cocon for Plixus Core	Plixus Core
GetSpeakSlowerRequestList	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get event form Interpreter desk for the speck slow button pressed.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetSpeakSlowerRequestList>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetSpeakSlowerRequestList>

**Return:**

```
{"SpeakSlowerRequest":{"UnitId":3}}
```

### 3.3.13.20 GetInterpreterPresetById

API Module	Cocon for Plixus Core	Plixus Core
GetInterpreterPresetById	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Get full information of interpreterpreset by id.

**Structure:**

<http://localhost:8890/CoCon/Interpretation/GetInterpreterPresetById/?Id=2>

**Example:**

<http://localhost:8890/CoCon/Interpretation/GetInterpreterPresetById/?Id=2>

**Return:**

```
{"GetInterpreterPresetById":{"InterpreterPreset":{"Id":12,"Name":"Test 100","Description":"XYZ","DeskMode":0,"BoothMode":0,"MultipleChannelActivation":false,"IsOverrideChannel":false,"IsAllActiveMicsLEDBlink":false,"AllActiveMicsLEDBlinkTime":5,"IsSlowDownImplementedExternally":true,"IsListenerCountConfigurator":false,"ListenerCountConfiguratorTime":60,"State":"Active","Channels":[{"Order":1,"Language":"Hindi","Booth":"1","Desks":[{"Name":"1","ActiveChannel":0,"Relays":[]}, {"Name":"2","ActiveChannel":0,"Relays":[]}],"IsRecording":false}, {"Order":2,"Language":"English","Booth":"2","Desks":[{"Name":"1","ActiveChannel":0,"Relays":[]}],"IsRecording":false}}}}
```

### 3.3.14 API calls for Configurations

#### 3.3.14.1 GetCulpAddress

API Module	Cocon for Plixus Core	Plixus Core
GetCulpAddress	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Get CU IP address from the Room server config file (which CU is currently connected).

**Structure:**

<http://localhost:8890/CoCon/GetCulpAddress>

**Return:**

```
{"GetCulpAddress":{"IpAddress":"192.168.8.110"}}
```

#### 3.3.14.2 SetCulpAddress

API Module	Cocon for Plixus Core	Plixus Core
SetCulpAddress	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Set / Update CU IP address to CoCon room server config file and for the taking impact of the modified IP address user should have to restart room server.

**Structure:**

<http://localhost:8890/CoCon/SetCulpAddress/?IpAddress=192.168.8.110>

**Return:**

```
{"SetCulpAddress":{"Result":true,"IpAddress":"192.168.8.110"}}
```

### 3.3.14.3 RestartRoomServer

API Module	Cocon for Plixus Core	Plixus Core
RestartRoomServer	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Restart Room server.

**Structure:**

<http://localhost:8890/CoCon/RestartRoomServer>

**Return:**

Simply 3rd party tool lost connection.

### 3.3.14.4 GetCoconServerVersion

API Module	Cocon for Plixus Core	Plixus Core
GetCoconServerVersion	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This API calls reply with the Cocon room server version.

**Structure:**

<http://localhost:8890/CoCon/GetCoconServerVersion>

**Return: string**

```
{"GetCoconServerVersion":{"Version":"5.3.0.52"}}
```

### 3.3.15 Recording ( AE-R & Plixus Core only)

#### 3.3.15.1 Getrecordingfilesinfo

API Module	Cocon for Plixus Core	Plixus Core_AE-R
GetRecordingFilesInfo	<input type="checkbox"/>	<input checked="" type="checkbox"/>

This api call we return of the recored files information from CU.

**Structure:**

<http://192.168.8.235:8890/CoCon/Recording/getrecordingfilesinfo>

**Return: string**

```
{"GetRecordingFilesInfo":{"RecordingFilesInfo":{"RecordingFiles":[{"Name":"/audio/internal/1970-01-01_00h04m36s_01-EN_00.mp3"}, {"Name":"/audio/internal/1970-01-01_00h04m36s_Floor_00.mp3"}, {"Name":"/audio/internal/1970-01-01_00h05m20s_01-EN_00.mp3"}, {"Name":"/audio/internal/1970-01-01_00h05m20s_Floor_00.mp3"}, {"Name":"/audio/internal/test_1970-01-01_00h06m01s_01-EN_Test_00.mp3"}, {"Name":"/audio/internal/test_1970-01-01_00h06m01s_Floor_Test_00.mp3"}, {"Name":"/audio/internal/test_1970-01-01_00h08m20s_01-EN_Test_00.mp3"}, {"Name":"/audio/internal/test_1970-01-01_00h08m20s_Floor_Test_00.mp3"}], "StorageType":"internal"}, {"RecordingFiles": [], "StorageType":"external"}]}
```

**3.3.15.2 Startrecording**

API Module	Cocon for Plixus Core	Plixus Core_AE-R
Startrecording	<input type="checkbox"/>	<input checked="" type="checkbox"/>

This api call will start recording and store the file based on the configuration.

NOTE1: Both these arguments are optional. If the arguments are not present ( eg: Recording/StartRecording ), the last set prefix and suffix are used. To clear the prefix and/or suffix the argument must be passed with "" (Prefix="")

If Succeeded: RecordingState: Active

When an invalid string is passed as Prefix or Suffix:

Errors: {RecordError: kErrorIllegalSuffix} and/or {RecordError: kErrorIllegalPrefix}

NOTE2: These ONLY appear in case of an error, if the strings are ok there is no Errors field.

NOTE3: Even if these appear, the recording will start. The recording file will simply omit the invalid prefix or suffix and an empty prefix or suffix.

NOTE4: This function will start recording the floor if no channels have been selected to record in a previous setup.

**Parameters:**

Name	Description	Type
Prefix	File name start with (Optional)	string
Suffix	File name end with (Optional)	string

**Structure:**

<http://192.168.8.235:8890/CoCon/Recording/Startrecording>

<http://192.168.8.235:8890/CoCon/Recording/Startrecording/?Prefix={prefix}&Suffix={suffix}>

**Example :**

<http://192.168.8.235:8890/CoCon/Recording/Startrecording/?Prefix=Room1&Suffix=Televic>

**Return: string**

```
{"StartRecording":{"RecordingState":"active"}}
```

**Update event :**

```
{"RecordingStateUpdated":{"RecordingState":"active"}}
```

### 3.3.15.3 Stoprecording

API Module	Cocon for Plixus Core	Plixus Core_AE-R
Stoprecording	<input type="checkbox"/>	<input checked="" type="checkbox"/>

This api call will stop the current recording.

**Structure:**

<http://192.168.8.235:8890/CoCon/Recording/Stoprecording>

**Return: string**

```
{"StopRecording":{"RecordingState":"idle"}}
```

**Update Event :**

```
{"RecordingStateUpdated":{"RecordingState":"idle"}}
```

### 3.3.15.4 Getrecordingstate

API Module	Cocon for Plixus Core	Plixus Core_AE-R
Getrecordingstate	<input type="checkbox"/>	<input checked="" type="checkbox"/>

This api will get the current state of the recording.

State can be:

active  
error  
idle  
paused  
unknown.

Note: There is also one Server to Client call whenever the RecordingState on the device is updated:

**Structure:**

<http://192.168.8.235:8890/CoCon/Recording/Getrecordingstate>

**Return: string**

```
{"GetRecordingState":{"RecordingState":"idle"}}
```

### 3.3.16 Wireless coupling

#### 3.3.16.1 Getwcapinfo

API Module	Cocon for Plixus Core	Plixus Core
Getwcapinfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This api call will get the information of the WCAP.

**Structure:**

[http://192.168.8.240:8890/CoCon/Wireless/Getwcapinfo/?WcapSerial={wcapSerial\\_number}](http://192.168.8.240:8890/CoCon/Wireless/Getwcapinfo/?WcapSerial={wcapSerial_number})

**Example :**

<http://192.168.8.240:8890/CoCon/Wireless/Getwcapinfo/?WcapSerial=352321544>

**Return: string**

```
{"WcapInfoUpdated":{"chans":[],"cpld":0,"gtw":"192.168.8.1","hn":"admin","ipa":"192.168.8.210","mac":"1C:BA:8C:99:FE:81","online":1,"op_mode":1,"reg":"Europe","sbm":"255.255.255.0","serial":352321544,"wlist":[]}}
```

### 3.3.16.2 Getallwcapinfo

API Module	Cocon for Plixus Core	Plixus Core
Getallwcapinfo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

This api call will get All WCAP information which is connected.

**Structure:**

<http://192.168.8.235:8890/CoCon/Wireless/Getallwcapinfo>

**Return: string**

```
{"AllWcapInfoUpdated":[{"chans":[],"cpld":1,"gtw":"192.168.8.1","hn":"admin","ipa":"192.168.8.210","mac":"1C:BA:8C:99:FE:81","online":1,"op_mode":1,"reg":"Europe","sbm":"255.255.255.0","serial":352321544,"wlist":[]}]}
```

### 3.3.16.3 SetWcapCoupledMode

API Module	Cocon for Plixus Core	Plixus Core
SetWcapCoupledMode	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set the WCAP coupling mode.

**Structure:**

<http://192.168.8.120:8890/CoCon/Wireless/SetWcapCoupledMode/?WcapSerial=352321544&CoupledMode=False>

**Return: int**

0 → operation success; 1 → Operation fail (not valid value) -1 → exception

**Update Event :**

```
{"WcapCoupledUpdated":{"CoupledMode":false,"WcapSerial":352321544}}
```

## 4 Future commands

### 4.1 Server to client

---

### 4.2 Client to server

---