1 Scope 6
2 References 7
3 Version update records 7
4 Definitions and abbreviations 9
4.1 Definitions 9
4.2 Abbreviations 10
5 Architecture and Transmission Mechanism 10
5.1 REST and HTTP Methods 10
5.2 XML 11
5.3 Resources overview 11
5.4 Protocol URL 11
5.5 Messages 12
5.5.1 Connection Header Field 12
5.5.2 Authorization and WWW-Authenticate Header Fields 12
5.5.3 Entity Body 13
5.5.4 Operations 13
6 Device discovery 15
7 Resource Description 15
7.1.1 Resource Description Outline 15
7.1.2 Annotation 16
8 Error Handling 16
9 Services and General Resources 20
9.1 /Network 20
9.1.1 /Network/PPPOE 20
9.1.1.1 /Network/PPPOE/Config 21
9.1.1.2 /Network/PPPOE/Status 22
9.1.2 /Network/DDNS 22
9.1.2.1 /Network/DDNS/ServerList 23
9.1.2.2 /Network/DDNS/Config 24
9.1.2.3 /Network/DDNS/Status 25
9.1.3 /Network/UPNP 25
9.1.3.1 /Network/UPNP/Config 26
9.1.3.2 /Network/UPNP/Status 27
9.1.4 /Network/Email 27
9.1.5 /Network/Port 29
9.1.6 /Network/P2P 30
9.1.7 /Network/P2PV2 30
9.1.8 /Network/FTP 31
9.1.9 /Network/WifiParamTest 32
9.1.10 /Network/WifiParamTestStatus 33
9.1.11 /Network/WhichNetworkCard 33
9.1.12 /Network/Interfaces 34
9.1.12.1 /Network/Interfaces/NetworkCardID 35
9.1.12.2 /Network/Interfaces/3/4GCardCfg 41
9.1.12.3 /Network/Interfaces/3/4GCardInfo 41
9.1.12.4 /Network/Interfaces/3/4GCardTest 42
9.1.13 /Network/CloudStorage 42
9.1.13.1 /Network/CloudStroage/BaseConfig 43
9.1.13.2 /Network/CloudStroage/OpenStatus 43
9.1.13.3 /Network/CloudStroage/RecordPlan 44
9.1.13.4 /Network/CloudStroage/BaseStatus 45
9.1.13.5 /Network/CloudStroage/BusinessInfo 45
9.2 /System 46
9.2.1 /System/Time 46
9.2.2 /System/NTP 47

9.2.3 /System/DST 47
9.2.4 /System/DeviceInfo 48
9.2.5 /System/DeviceStatus 49
9.2.5.1 /System/DeviceStatus/ChannelID 50
9.2.6 /System/FirmwareUpgrade 51
9.2.7 /System/Reboot 51
9.2.8 /System/FactoryDefault 51
9.2.9 /System/FactoryDefaultV2 52
9.2.10 /System/SendTestEmail 52
9.2.11 /System/RecoveryAdminPassword 52
9.2.12 /System/TryListenCurrentAlarmAudio 53
9.2.13 /System/Logging 53
9.2.14 /System/Users 55
9.2.15 /System/Users/ID 56
9.2.16 /System/ChannelID/RemoteForceIFrame 57
9.2.17 /System/ChannelID/RemoteAlarmOutClear/ID 57
9.2.18 /System/ChannelID/RemoteAlarmoutControl/on 58
9.2.19 /System/ChannelID/RemoteAlarmoutControl/off 58
9.2.20 /System/AlarmoutState 58
9.2.21 /System/AlarmInState 59
9.2.22 /System/DeviceConfigFile 59
9.2.23 /System/TransparentChannel/WriteOnce 60
9.2.24 /System/TransparentChannel/Connect 60
9.2.25 /System/TransparentChannel/Write 61
9.2.26 /System/TransparentChannel/Read 62
9.2.27 /System/TransparentChannel/Disconnect 62
9.2.28 /System/DeviceAutoMaintenance 63
9.2.29 /System/SendTestFTP 63
9.2.30 /System/DeviceWifiSTA2AP 64
9.2.31 /System/DeviceWifiAP2STA 64
9.2.32 /System/DeviceCustomFunction 64
9.2.33 /System/CloudUpgradeServerInfo 65
9.2.34 /System/CloudUpgradeFirmwareList 65
9.2.35 /System/CloudUpgradeDownloadStart 66
9.2.36 /System/CloudUpgradeDownloadStatus 66
9.2.37 /System/CloudUpgradeDownloadTerminate 67
9.2.38 /System/CloudUpgradeStartUpgrade 67
9.2.39 /System/CloudUpgradeResult 68
9.2.40 /System/ErrorInfoMap 68
9.2.41 /System/DeviceCap 69
9.2.42 /System/AudioAlarmConfig 70
9.2.43 /System/TryListenSelectedAlarmAudio 71
9.2.44 /System/SaveHardwareConfig 71
9.2.45 /System/CancelHardwareConfig 71
9.2.45 /System/AuthInfo 72
9.2.46 /System/DeviceCMS 72
9.2.47 /System/DeviceInfoV2 73
9.2.48 /System/DeviceReversePacket 73
9.2.49 /System/ChannelID/AssistantInterfaceControl 74
9.2.50 /System/WorkModeControl 74
9.2.51 /System/DeviceBindConfig 75
9.2.52 /System/LongSeeDeviceConfig 75
9.2.53 /System/DeleteVoiceCustom 76
9.2.54 /System/Language 76
9.2.55 /System/PerformanceStatus 76

9.2.56 /System/FiberHomeBaseInfo 77
9.2.57 /System/FiberHomeLocationInfo 77
9.2.58 /System/FiberHomeCtrlChnlServer 78
9.2.59 /System/FiberHomeLocationID 78
9.2.60 /System/FiberHomeDataCenterServer 79
9.2.61 /System/FiberHomeAreaCodeInfo 79
9.2.62 /System/MQTTServerConfig 80
9.2.63 /System/AIWorkMode 81
9.3 /Security 81
9.3.1 /Security/UserAuth 81
9.3.2 /Security/ModifyPasswordByTempSecretKey 82
9.3.3 /Security/BaseConfig 82
9.3.4 /Security/WhiteLists 82
9.3.5 /Security/WhiteList 83
9.3.6 /Security/UserPermissionConfigs 85
9.3.7 /Security/UserPermissionConfig/UserID 86
9.3.8 /Security/UserPermissionConfigByUsername 86
9.4 /Streams 87
9.4.1 /Streams/ChannelID 88
9.4.1.1 /Streams/ChannelID/ReceiveData 88
9.4.1.2 /Streams/ChannelID/CapabilityV2 89
9.4.1.3 /Streams/ChannelID/StreamID 90
9.4.2 /Streams/AudioStream 92
9.5 /Snapshot 93
9.5.1 /Snapshot/ChannelID/RemoteImageCapture 93
9.5.2 /Snapshot/ChannelID/StreamID/RemoteImageCaptureV2 93
9.5.3 /Snapshot/Capability 94
9.5.4 /Snapshot/Config 94
9.6 /Images 95
9.6.1 /Images/ChannelID 96
9.6.1.1 /Images/ChannelID/Scene 96
9.6.1.2 /Images/ChannelID/Basic 97
9.6.1.3 /Images/ChannelID/Color 98
9.6.1.4 /Images/ChannelID/Exposure 99
9.6.1.5 /Images/ChannelID/WhiteBalance 99
9.6.1.6 /Images/ChannelID/IrCutFilter 100
9.7 /Pictures 102
9.7.1 /Pictures/ChannelID 103
9.7.1.1 /Pictures/ChannelID/OSD 103
9.7.1.2 /Pictures/ChannelID/MultiOSD 104
9.7.1.3 /Pictures/ChannelID/MultiOSDV2 105
9.7.1.4 /Pictures/ChannelID/VideoLost 106
9.7.1.5 /Pictures/ChannelID/Motion 110
9.7.1.6 /Pictures/ChannelID/PeopleDetect 115
9.7.1.7 /Pictures/ChannelID/PeopleDetect/Ability 118
9.7.1.8 /Pictures/ChannelID/FaceDetect 118
9.7.1.9 /Pictures/ChannelID/Mosaic 120
9.7.1.10 /Pictures/ChannalID/LightFillingMode 121
9.7.1.11/Pictures/ChannalID/TraversePlaneDetect 122
9.7.1.12/Pictures/ChannalID/FieldDetect 123
9.7.1.13/Pictures/ChannalID/PIRDetect 126
9.8 /PTZ 128
9.8.1 /PTZ/ChannelID/Presets/Set 128
9.8.2 /PTZ/ChannelID/Presets/Remove 128
9.8.3 /PTZ/ChannelID/Presets/Goto 129

9.8.4 /PTZ/ChannelID/Cruise/StartCruiseMem 129
9.8.5 /PTZ/ChannelID/Cruise/StopCruiseMem 130
9.8.6 /PTZ/ChannelID/Cruise/ClearCruise 130
9.8.7 /PTZ/ChannelID/Cruise/AddPresetToCruise 131
9.8.8 /PTZ/ChannelID/Cruise/StartCruise 132
9.8.9 /PTZ/ChannelID/Cruise/StopCruise 132
9.8.10 /PTZ/ChannelID/Track/StartTrackMem 133
9.8.11 /PTZ/ChannelID/Track/StopTrackMem 133
9.8.12 /PTZ/ChannelID/Track/StartTrack 134
9.8.13 /PTZ/ChannelID/Track/StopTrack 134
9.8.14 /PTZ/ChannelID/FocusFar 135
9.8.15 /PTZ/ChannelID/FocusNear 135
9.8.16 /PTZ/ChannelID/ApertureBigger 136
9.8.17 /PTZ/ChannelID/ApertureSmaller 136
9.8.18 /PTZ/ChannelID/ZoomIn 137
9.8.19 /PTZ/ChannelID/ZoomOut 138
9.8.20 /PTZ/ChannelID/TurnLeft 138
9.8.21 /PTZ/ChannelID/TurnRight 139
9.8.22 /PTZ/ChannelID/TurnUp 139
9.8.23 /PTZ/ChannelID/TurnDown 140
9.8.24 /PTZ/ChannelID/TurnUpLeft 140
9.8.25 /PTZ/ChannelID/TurnUpRight 141
9.8.26 /PTZ/ChannelID/TurnDownLeft 141
9.8.27 /PTZ/ChannelID/TurnDownRight 142
9.8.28 /PTZ/ChannelID/AutoLeftRight 143
9.8.29 /PTZ/ChannelID/Auxiliary 143
9.8.30 /PTZ/ChannelID/Light 144
9.8.31 /PTZ/ChannelID/RainBrush 144
9.8.32 /PTZ/ChannelID/Fan 145
9.8.33 /PTZ/ChannelID/Hearter 145
9.8.34 /PTZ/ChannelID/Config 146
9.8.35 /PTZ/ChannelID/AdvanceConfig 147
9.8.36 /PTZ/ChannelID/Watch/StartWatch 148
9.8.37 /PTZ/ChannelID/Watch/StopWatch 148
9.8.38 /PTZ/ChannelID/HumanTrack/StartHumanTrack 149
9.8.39 /PTZ/ChannelID/HumanTrack/StopHumanTrack 149
9.8.40 /PTZ/ChannelID/Status 150
9.8.41 /PTZ/Config 150
9.9 /TwowayAudio 151
9.9.1 /TwowayAudio/Open 151
9.9.2 /TwowayAudio/Close 152
9.9.3 /TwowayAudio/SendData 152
9.9.4 /TwowayAudio/ReceiveData 152
9.10 /EventsNotification 153
9.10.1 /EventsNotification/SubscribedEventTypeList 153
9.10.2 /EventsNotification/Notify 154
9.10.3 /EventsNotification/EventList 155
9.11 /Alarm 155
9.11.1 /Alarm/AlarmIn 156
9.11.1.1 /Alarm/AlarmIn/ChannelID 156
9.11.2 /Alarm/AlarmOut 158
9.11.2.1 /Alarm/AlarmOut/ChannelID 158
9.11.3 /Alarm/AMS 160
9.11.4 /Alarm/AMS/1 160
9.11.5 /Alarm/AMS/2 161

9.11.6 /Alarm/EventCfgList 161
9.11.7 /Alarm/EventCfgList/1 162
9.11.8 /Alarm/EventCfgList/2 163
9.11.9 /Alarm/AlarmAction/AlarmID/LightWarning 164
9.11.10 /Alarm/AlarmAction/AlarmID/LightWarningByCurrentConfig 164
9.11.11 /Alarm/PreventionControl 164
9.12 /Record 165
9.12.1 /Record/HttpEvent/Config 166
9.12.2 /Record/HttpEvent/Call 166
9.12.3 /Record/Format/Call 166
9.12.4 /Record/Notify/URL 167
9.12.5 /Record/ChannelID/RecordSchedule 167
9.13 /Disk 168
9.14 /Playback 169
9.14.1 /Playback/GetRecordInfoByMonth 169
9.14.2 /Playback/GetRecordInfoByDay 170
9.15 RTMP 170
9.15.1 /RTMP/BasicConfig 171
9.15.2 /RTMP/ChannelID/StreamID/Config 171
9.16 /RecordFileList 172
9.16.1 /RecordFileList/Attribute 172
9.16.2 /RecordFileList/Pages/PageNum 173
9.17 /SnapshotFileList 173
9.17.1 /SnapshotFileList/Attribute 174
9.17.2 /SnapshotFileList/Pages/PageNum 174
9.18 /FaceGroup 175
9.18.1 /FaceGroup/Create 175
9.18.2 /FaceGroup/Modify 175
9.18.3 /FaceGroup/Delete 176
9.18.4 /FaceGroup/QueryAll 176
9.18.5 /FaceGroups/QueryByGroupID 177
9.18.6 /FaceGroups/QueryWorkState 178
9.18.7 /FaceGroup/CheckPersonInfo 178
9.18.9 /FaceGroup/AddPersonInfoAndFaceImage 179
9.18.10 /FaceGroup/DeletePersonList 182
9.18.11 /FaceGroup/UnBindPersonInfo 182
9.18.12 /FaceGroup/BindPersonInfo 183
9.18.13 /FaceGroup/ModifyPersonInfo 183
9.18.14 /FaceGroup/UpdateFaceImageInfoByPerson 184
9.18.15 /FaceGroup/UpdatePersonInfoAndFaceImage 186
9.18.16 /FaceGroup/QueryPersonInfoListCount 189
9.18.17 /FaceGroup/QueryPersonInfoList 189
9.18.18 /FaceGroup/QueryPersonImageInfo 191
9.18.19 /FaceGroup/QueryPersonInfoByPersonID 191
9.18.20 /FaceGroup/ImagePersonMap 192
9.18.21 /FaceGroup/RemoveImagePersonMap 192
9.18.22 /FaceGroup/ChannelID/FaceBrightness 193
9.19 /FaceReco 194
9.19.1 /FaceReco/ChannelID/RecoRuleList 194
9.19.2 /FaceReco/ChannelID/RecoRuleList 196
9.19.3 /FaceRecognition/QueryRecordCount 199
9.19.4 /FaceRecognition/QueryRecordList 199
9.19.5 /FaceRecognition/DetailByRecognitionRecord 200
9.19.6 /FaceRecognition/SnapshotByRecognitionRecord 201
9.19.7 /FaceRecognition/RecoResultImageByRecognitionRecord 201

9.19.8 /FaceRecognition/Ability 202
9.19.9 /FaceRecognition/FaceLivenessStatus 203
9.19.10 /FaceReco/ChannelID/BaseConfig 203
9.20 /Face 204
9.20.1 /Face/ChannelID/DefaultFaceRegionInfo 204
9.20.2 /Face/ChannelID/DetectRegion 205
9.20.3 /Face/ChannelID/Config 205
9.20.4 /Face/ChannelID/OverlayInfo 207
9.20.5 /Face/ChannelID/ClearFaceOverlayInfo 207
9.20.6 /Face/ChannelID/DefaultFaceRegionInfo 208
9.20.7 /Face/ChannelID/FixedRegionDetect 208
9.21 /AccessGate 209
9.21.1 /AccessGate/AddCard 209
9.21.2 /AccessGate/DeleteCard 210
9.21.3 /AccessGate/ModifyCard 210
9.21.4 /AccessGate/QueryAllCard 211
9.21.5 /AccessGate/AddPassword 211
9.21.6 /AccessGate/DeletePassword 212
9.21.8 /AccessGate/ModifyPassword 212
9.21.9 /AccessGate/QueryAllPassword 213
9.21.10 /AccessGate/QueryCardNumberByID?ID=xxx 213
9.21.11 /AccessGate/QueryPasswordByID?ID=xxx 213
9.22 /FaceMaskDetect 214
9.22.1 /FaceMaskDetect/AlarmConfig 214
9.23 /FaceTemperatureDetect 215
9.23.1 /FaceTemperatureDetect/ChannelID/FaceTemperatureDetectConfig 215
9.23.2 /FaceTemperatureDetect/ChannelID/OverlayFaceTemperatureOnOSD 216
9.23.3 /FaceTemperatureDetect/ChannelID/FaceTemperatureStatus 217
9.23.4 /FaceTemperatureDetect/ChannelID/FaceTemperatureAbnormalAlarmConfig 217
9.24 /Common 218
9.24.1 /Common/DeviceFileDownload 218
9.24.2 /Common/DeviceFileUpload 218
10 Device discovery V2 219

1 Scope

This specification defines a HTTP-based application programming interface that enables physical security and video management systems to communicate with IP media devices in a particular way.

With regard to Media Streaming, please refer to —develop API of RTSP protocol‖.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] RFC2616 Hypertext Transfer Protocol-HTTP/1.1

[2] W3C XML 1.0 specification

[3] W3C Character encodings

[4] RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax and Semantics

[5] RFC 2617 HTTP Authentication:Basic and Digest Access Authentication

[6] International Electrotechnical Commission —ISO/IEC standard on UPnP device architecture makes networking simple and easy‖, 2008-12-09. Retrieved on 2009-05-07.

[7] International Organization for Standardization —ISO/IEC standard on UPnP device architecture makes networking simple and easy‖, 2008-12-10. Retrieved on 2009-05-07.

[8] UPnP Forum —UPnP Specifications Named International Standard for Device Interoperability for IP-based Network Devices‖, 2009-02-05. Retrieved on 2009-05-07.

3 Version update records

Modify Date	Version	Modification
2016/04/22	1.1.9	New,first published
2016/05/31	1.1.10	1.Add the url relation with alarm. 2.Add the url relation with record. 3.Add the section “StorageDirectory” of the ftp. 4.Add the url “/Streams/ChannelID/ReceiveData”. 5.Add the url “/Streams/ChannelID/StreamID/ReceiveData”
2016/10/08	1.1.11	Add the URL “/Network/WifiParamTest”
2016/10/14	1.1.12	1.Add the url “/Network/WifiParamTestStatus”. 2.Add the url “/Network/WhichNetworkCard”.
2016/11/01	1.1.13	1.Add the url “/Record/Notify/URL”. 2.Add the url”/Record/ChannelID/RecordSchedule”
2016/11/08	1.1.14	Add the url “/Pictures/ChannelID/Motion/RegionV2”
2016/11/14	1.1.15	1.Add the url “/Playback/GetRecordInfoByMonth”

		2.Add the url “/Playback/GetRecordInfoByDay”
2016/12/21	1.1.15	1.Update the url “/Network/Email”. 2.Update the url “/Network/FTP”. 3.Update the url “/Images/ChannelID/Scene”. 4.Update the url “/Images/ChannelID/Exposure” 5.Update the url “System/NTP”
2017/01/09	1.1.17	Add the url “/System/SendTestFTP”
2017/01/17	1.1.18	1.Add the url “System/DeviceWifiSTA2AP”. 2.Update the url “/System/DeviceWifiSTA2AP”. 3.Update the url “/Network”. 4.Update the url “Streams/AudioStream”. 5.Update the url “Record”.
2017/02/14	1.1.19	Add the url “/System/FactoryDefaultV2”
2017/04/14	1.1.20	1.Modify the error position. 2.Add the url “/System/DeviceCustomFunction”. 3.Add the url “/Network/P2PV2”.
2017/05/04	1.1.21	1.Updata the url /Record 2.Update the url /Record/ChannelID/RecordSchedule
2017/06/28	1.1.22	1,Add a new field for WPA password encoded by Base64.
2017/07/12	1.1.23	1.Add URL “RTMP/BasicConfig” for RTMP basic configuration. 2.Add URL “/RTMP/ChannelID/StreamID/Config” for RTMP stream configuration. 3.Add RTMP capability filed in URL “/System/DeviceInfo”.
2017/07/27	1.2.0	1,Add h265+ capability and codec setting.
2017/08/04	1.2.1	1.Add the url “/Pictures/ChannelID/Motion/Notify/URL”
2017/08/11	1.3.0	Add h264+ capability and codec setting.
2017/11/10	1.3.1	1.Add the url “/Alarm/AMS” 2.Add the url “Alarm/EventCfgList”
2018/05/04	1.3.2	1. Add the the volume of audio out(/Streams/AudioStream). 2. Add the url “/System/CloudUpgradeServerInfo” 3. Add the url “/System/CloudUpgradeFirmwareList” 4. Add the url ”/System/CloudUpgradeDownloadStart” 5. Add the url “/System/CloudUpgradeDownloadTerminate” 6. Add the url “/System/CloudUpgradeDownloadStatus” 7. Add the url “/System/CloudUpgradeStartUpgrade” 8. Add the url “/System/CloudUpgradeResult” 9. Add the smart color function(/Images/ChannelID/IrCutFilter). 10. Add the url “/RecordFileList/Attribute” 11. Add the url “/RecordFileList/Pages/(Pagenum)” 12. Add the url “/System/ErrorInfoMap” 13. Add the url “/System/DeviceCap” 14. Old version only support for sensitivity of highest,higher,lower and lowest,some version support for sensitivity of highest, higher,high, middle,low, lower and lowest(/Pictures/1/Motion). 15. Modified the problem of getting the failure of the MJPEG stream(/Streams/ChannelID/StreamTypeID/ReceiveData). 16. support for half time zone or 1/4(/System/NTP). 17. Add the url “/SnapshotFileList/Attribute” 18. Add the url “/SnapshotFileList/Pages/(PageNum)”
2018/05/30	1.3.3	1. add the status “nodisk,exceptionneedformat,exceptionneedChange Disk,formatfailedneedreboot” int the <DiskStorageAttribute> of the url /Disk 2. add the status “umount failure,device is busy” int the <DiskFormatStatus> of the url /Disk 3. add the status “passwd error” int the

		<Status> of the url /Network/WifiParamTestStatus 4. Add the url /Snapshot/ChannelID/StreamID/RemoteImageCaptureV2 5. Modify the url /RecordFileList/Pages/PageNum. 6. Modify the url /SnapshotFileList/Pages/PageNum.
2018/07/10	2.0.0	1. Add the url "/PTZ/ChannelID/TurnUpLeft" 2. Add the url "/PTZ/ChannelID/TurnUpRight" 3. Add the url "/PTZ/ChannelID/TurnDownLeft" 4. Add the url "/PTZ/ChannelID/TurnDownRight" 5. Add the url "/PTZ/ChannelID/AutoLeftRight"
2019/08/14	2.0.6	1. Add the url "/Network/Interfaces/2/WIFIAccessPointListV2"
2019/10/22	1.4.3	1.Add the url" /FaceGroup/ImagePersonMap" 2.Add the url"/PTZ/ChannelID/HumanTrack/StartHumanTrack" 3.Add the url"/PTZ/ChannelID/HumanTrack/StartHumanTrack"
2019/10/23	1.4.4	1.Add "ePTZControlCodeTrackRunStart" and "ePTZControlCodeTrackRunStop" in ePTZControlCmdCodem(enum) 2.Add field "AI-265" and "AiFace" in HY_DVR_PEOPLE_DETECT(struct)
2019/10/26	1.4.5	1.Add the host filed in "HY_DVR_CLOUD_STORAGE" 2.Add the url "/Network/CloudStorage/BaseConfig" 3.Add the url "/Network/CloudStorage/OpenStatus" 4.Add the url "/Network/CloudStorage/RecordPlan" 5.Add the url "/Network/CloudStorage/BaseStatus" 6.Add the url "/Network/CloudStorage/BusinessInfo" 7.Add the url "/FaceGroup/RemoveImagePersonMap"
2019/11/04	1.4.6	1.Add the url "/Alarm/AlarmOut/ChannelID/ControlMode" 2.Add the url "/Pictures/ChannelID/LightFillingMode" 3.Add the url "/FaceGroup/ChannelID/FaceBrightness" 4.Add the url "/Alarm/AlarmAction/AlarmID/LightWarning"
2019/11/14	1.4.7	1.Add the url "/Alarm/AlarmAction/ChannelID/LightWarningByCurrentConfig"
2019/11/22	1.4.8	1.Add the url "/PTZ/ChannelID/Status" 1.Add the url "/Face/ChannelID/DefaultFaceRegionInfo"
2019/11/27	1.4.9	1.Add the url "/Pictures/ChannelID/PeopleDetect/Ability" 2.Add the url "/FaceRecognition/FaceLivenessStatus"
2019/12/03	1.5.0	1.Add the url "/Common/DeviceFileDownload" 2.Add the url "/Common/DeviceFileUpload"
2019/12/04	1.5.1	1.Add field "nVarInfraredWorkMode","nVarInfraredBrightness" in ZW_IPC_ISP_EX
2019/12/11	1.5.2	1.The url "/Images/channelID/IrCutFilter" get xml add two field "InitiveSwitchTime"," PassivitySwitchTime" 2.return the sensitivity info in any mode
2019/12/19	1.5.3	1.Add "HKPort" in the return xml of url "/Network/Port"
2019/12/24	1.5.4	1.fix the problem that the cloud storage plan can not set correctly. 2."/Network/CloudStorage/BaseStatus" Get xml add "ErrorCode" field.

4 Definitions and abbreviations

4.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Services: a set of resources consisting of relevant General Resources.

General Resources: physical resources that supported by the devices.
Node: Services and General Resources.

4.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:
REST REpresentational State Transfer
UPnP Universal Plug and Play

5 Architecture and Transmission Mechanism

The IP Media Device Management Protocol is based on REST architecture. The management and control interfaces defined in this specification are treated as resources utilizing the REpresentational State Transfer (REST) architecture. This architecture facilitates users by grouping related resources within hierarchical namespaces, and is more flexible for service discovery and future expansion.

REST architecture consists of clients and servers, among which clients initiate request to servers, while servers handle requests and response accordingly. Requests and responses are established via the transmission of ―representations‖ of ―resources‖. REST architecture need to be based on an Application Layer protocol which provides various of standard communication formats for applications based on the transfer of meaningful representational state. HTTP[1] has a very rich vocabulary in terms of verbs(or ―methods‖),URIs, request and response headers, Internet media types, HTTP request and response codes etc. In addition, HTTP also has some features particularly suitable for REST architecture. So HTTP is used as external Application Layer protocol in this specification. In the architecture, clients are physical security and video management systems; servers are IP media devices.

This specification also contains full XML schema for the introduced resources.

5.1 REST and HTTP Methods

The following table shows how HTTP verbs are typically used to implement a web service based on REST architecture.

Resourc	GET	PUT	POST	DELETE

Collection URI, such as http://webServer/resour ces	**List** the members of collection,complete with their member URIs for further navigation.	Meaning defined as —**replace** the entire collection with another collection‖.	**Create** a new entry in the collection where the ID is assigned automatically by the collection. The ID created is usually included as part of the data returned by this	Meaning defined as —**delete** the entire collection‖.
Member URI, such as http://webServer/resour ces/7416	**Retrieve** a representation ofthe addressed member of the collection expressed in an appropriate MIME type.	**Update** the addressed member of the collection or **create** it with the specified ID.	Treat the addressed member as a collection in its own right and **create** a new subordinate of it.	**Delete** the addressed member of the collection.

5.2 XML

A device must support the syntax defined by W3C XML 1.0 specification [2] and UTF-8 character set [3]. All XML files must adopt UTF-8 encoding according to RFC3629. Additionally, all resources share a common XML schema as defined in Annex.

Any resources can specify separate input and output XML Documents. If a specific data structure is defined inside these documents, then they must be specified as XML Schema Documents (xsd) in Annex.

Lists contained in XML blocks will be represented in the format of <XXXList>, and each <XXXList> tag may contain one or more nodes.

5.3 Resources overview

Services	Description	Reference
Network	Configure network interfaces.	8.1
System	Configure and operate the general system functions.	8.2
Streams	Handle video-related configuration.	8.3
Images	Configure the general Images configuration	8.4
Pictures	Configure the general pictures configuration	8.5

5.4 Protocol URL

The URL scheme is used to locate device resources via a specific protocol in the network. This section defines the syntax and semantics for http(s) URLs.

protocol: URL scheme for the particular request. The http and https protocols are allowed in this specification.

host: The host field refer to the hostname, IP address, or the FQDN of an IP device.

port: The port field refer to the port number of that host on which the identified resource is located at the IP device listening for TCP connections. If the port is empty or not given, the default port is assumed. For HTTP, the default port 80. For HTTPS, the default port 443.

abs_path: The Request-URI [1] for the resources is abs_path [4]. The abs_path in this specification is most often of the form ―[/Services][/General Resources][/Special Resources]‖, which is suitable for resources to update or restore device configurations. ―*ID*‖ which appears in the abs_path identifies one resource of a list resource in this specification.

query: The query field is a string of information to be interpreted by the resource. It can include some resource-related parameters. It must be listed in name-value pair syntax (p1=v1&p2=v2&…&pn=vn). Each resource can define a set of parameters. Defining input data which is specific to the resource will be prior than query usage.

5.5 Messages

HTTP messages are used for communication between physical security and video management systems and IP media devices in this specification. In order to configure and control the device, some provisions are specified for these HTTP message.

5.5.1 Connection Header Field

Devices that implement HTTP/1.1 should support persistent connections in order to meet video management systems or client applications' requirements that issue multiple HTTP(s) transactions. HTTP/1.1 is implemented and utilized according to RFC 2616 in the IP devices. For a video management system or client application that uses persistent connection for multiple transactions, it is required to implement ―Connection: Keep-Alive‖ HTTP header field, while also adopt the ―Connection: close‖ HTTP header field for the last transaction of the persistent connection. This process will assume that the application can identify the last request in a sequence of multiple requests.

5.5.2 Authorization and WWW-Authenticate Header Fields

When a video management system or client application sends any request to the device, it must be authenticated by means of Basic Access [5] according to RFC 2617, and thus all the devices are required to support Basic Access. Authorization header field is sent along with each request, and if a user is authenticated, the request will follow the normal execution flow. If client HTTP request is with no authentication credentials,

unauthorized HTTP response (401) will be returned with WWW-Authenticate header field.

5.5.3 Entity Body

The Content-Type entity-header field indicates the media type of the entity body. The Content-Type may be designated as —application/xml;charset='UTF-8'‖, —application/octet-stream‖, etc.

For configuration information, the Content-Type is usually—application/xml; charset='UTF-8'‖. For example,

HTTP Request Message

```
GET /System/status HTTP/1.1
…
```

HTTP Response Message

```
HTTP/1.1 200 OK
…

Content-Type: application/xml; charset=‖UTF-8‖
…
```

HTTP Request Message

```
PUT /System/configurationData HTTP/1.1
…
Content-Type: application/octet-stream
…
[proprietary configuration file data content ]
```

HTTP Response Message

```
HTTP/1.1 200 OK
…

Content-Type: application/xml; charset=‖UTF-8‖
…
<?xml version=‖1.0‖ encoding=‖UTF-8‖?>
<ResponseStatus>
…
</ResponseStatus>
```

5.5.4 Operations

Different resources will specify different operation.

- The set device configuration‖ resources use PUT operation. If there is an XML block parameter for the request, the inbound XML format is defined according to a resource-special XML schema. Request status will be returned by the XML response information of the device, and can be used for indicating the PUT operation status. The responded XML format is defined by ―XML Response Schema‖ (please refer to section 4.5.5 for detail description). After the device configuration is updated successfully, it will return an XML response with status code ―OK‖; while another status code will be used for indicating unsuccessful operations. In either case, the device only responses after it is ready to continue normal operation, i.e. accepting streaming request, receiving configuration commands, etc.
- The get device configuration‖ resources use GET operation. After a successful GET operation, the result will be returned in XML format according to the resource description. For an unsuccessful request (i.e. users is not authenticated), the result will be returned in XML format according to ―XML Response Schema‖.
- Resources to create device configurations information will use the POST operation. If there is an XML block parameter for the request, the inbound XML format is defined according to a resource-special XML schema. The request status will be indicated by the XML response information returned from the device, and can be used to indicate the status of the POST operation. This XML format is defined according to ―XML Response Schema‖ (see section 4.5.5 for details). After successfully creating the data, the device returns an XML response with status code ―OK‖. A separate status code is used for unsuccessful operations.
- Resources to delete device configurations information will use the Delete operation. If successful, the result will be returned an XML response with status code ―OK‖. A separate status code is used for unsuccessful operations. This XML format is defined according to ―XML Response Schema‖ (see section 4.5.5 for details).
- Data uploading resources (i.e. firmware upgrade, import configuration, etc.) will use PUT operation. The content of the data will be stored in the body of the HTTP request. If successful, the result will be returned an XML response with status code ―OK‖. A separate status code is used for unsuccessful operations. This XML format is defined according to ―XML Response Schema‖ (see section 4.5.5 for details).
- Data receiving resources (i.e. export configuration file) use GET operation. If successful, the result will be returned the data according to the resource description. An XML block is used for unsuccessful operations. This XML format is defined according to ―XML Response Schema‖ (see section 4.5.5 for details).
- For Special Resources, GET operation will be used. For more detailed description see Section 6.

If there is an XML block for the HTTP request or response, the Content-Type and Content-Length will be set in the headers of the HTTP message.

6 Device discovery

The IP devices support Universal Plug and Play (UPnP) technology to discovery/locate themselves. A UPnP compatible device will automatically announce its network address, supported devices and services types when connected to a network, and therefore becoming ―plug-and-play‖ by allowing clients recognize those information and begin using this device immediately.if you want to get more information about upnp,please refer to《Discovery version 1.0.0.docx》

7 Resource Description

7.1.1 Resource Description Outline

Each resource in this specification is defined using the following format.

Resource_URI	***Type Version***
Operation_Name	***User Lever***
Description	*Description of the operation.*
Query	*Indicates the name/value pairs (p1, p2, p3,…,pn) for the resource.*
Inbound Data	*Indicates inbound data for the resources.*
Success Return	*the Type (if present) and the name of XML Data Block*
Notes: d*escribes any special processing rules for the resource.*	

Type refers to ―Special Resource‖, ―Service‖ and ―General Resource‖.

Version is used to determine the version of the protocol. The version number shall be set to ―1.0‖ in this specification.

Operation_Name refers to ―GET‖, ―PUT‖, ―POST‖ and ―DELETE‖.

Inbound Data includes three types as follows:

- NONE –no input data
- DataBlock – the name of an XML Data Block. Datablocks used here must be defined according to the specification.
- Mime type – mime type for the input data in the HTTP payload. Remark:―application/ xml‖ is not a valid mime type.

If a device does not support particular XML tags or blocks, then it may not be supported by the resource operations.

Generally, if a field is not provided in the inbound XML, then its current values shall not be

modified in the device's repository.

If a required field did not exist in the device's repository, then it must be provided in the applicable resource operations.

Success Return and Error Return detailed description see Section 8.

7.1.2 Annotation

The XML Data Blocks described in this document contains annotations for the field's properties. Please refer to the XML schema definitions for detail description。

The Following annotation content is inserted into the comments to describe the data carried in the field.

Table 6

Annotation	Description
req	Required field.
opt	Optional field. For data uploaded to the device, if the field is present but the device does not support it, it should be ignored.
dep	This field is required depending on the value of another field.
ro	Read-only. For XML data that is both read and written to the device, this field is only present in XML returned from the device. If this field is present in XML uploaded to the device, it should be ignored.
wo	Write-only. This field is only present in XML that can be uploaded to the device. This field should never be present in data returned from the device. [This is used for uploading passwords].
xs:<type>	A type defined in XML Schema Part 2: Datatypes Second Edition, see http://www.w3.org/TR/xmlschema-2

Remark: optional XML structures may contain required fields for the operation, which mean that even if the entire XML block is optional, some of its contained fields may still be necessary if required.

8 Error Handling

As with any other protocol, errors may occur during communications, protocol or message processing, and the specification classifies error handling into categories below:

- Protocol Errors, which are result of an incorrectly formed protocol message. Protocol Errors may contain header value or be received in an not expected or experience a socket timeout. To indicate and interpret protocol error, HTTP

protocol has defined a set of standard status codes [e.g., 1xx, 2xx, 3xx, 4xx, 5xx]. According to this specification, the IP devices will use appropriate HTTP protocol defined status codes for error reporting and when received handle accordingly.

- Application Errors, which are generated as a result of REST operations errors. All such application errors must be reported and handled through HTTP messages. The following table indicates the mapping relationship between HTTP status codes and REST operations, and also the information contained in response header and bodies.

HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
200	―OK‖-The request has succeeded. Header Notes: None Body Notes: The requested resource will be returned in the body.	√	√		√
201	―Created‖- The request has created a new resource. Header Notes: The Location header contains the URI of the newly created resource. Body Notes: The response returns an entity describing the newly created resource.		√	√	
204	―No Content‖ – The request succeeded, but there is no data to return. Header Notes: None Body Notes: No body is allowed.		√		√
301	―Moved Permanently‖ – The requested resource has moved permanently. Header Notes: The Location Header contains the URI of the new location. Body Notes: The body may contain the new resource	√			

302	—Found‖ – The requested resource should be accessed through this location, but the resource actually lives at another location. This is typically used to set up an alias.	√			
HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
	Header Notes: The Location header contains the URI of the resource. Body Notes: The body may contain the new resource				
	—Bad Request‖ – The request was badly formed. This is commonly used for creating or updating a resource, but the data was incomplete or incorrect. Header Notes: The				
	—Unauthorized‖ – The request requires user authentication to access this resource. If the request contains invalid authentication data, this code is sent.				
403	—Forbidden‖ – The request is not allowed because the server is refusing to fill the request. A header suggests to the client when to try resubmitting the	√	√	√	√

When setting the configuration, it will return a XML with an error code。Such as:

ResponseStatus XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<ResponseStatus Version="1.0" xmlns="http://www.zwcloud.wang/ver10/XMLSchema">
    <requestURL>/Network/Interfaces/2</requestURL>
    <statusCode>0</statusCode>
</ResponseStatus>
```

<requestURL> indicates URI operation。
<statusCode> represents the return value by the device。

StatusCode	Description
0	Success
-1	Decode error,maybe the XML and URL do not correspond.
-2	Decode error, maybe the XML missing fields with <req> tag.
-3	Unsupported operation
-4	Decode error，maybe It is associated with the channel
-5	Parameter error
-6	Query log busy
-7	Query log error
-8	call SDK fails
-9	Memory overflow
-10	URL Error
-11	User OverFlow
-12	Invalid Operate
-13	User Is Exist
4097	Picture format is wrong
4098	Face does not exist in the picture
4099	There are multiple faces in the picture
4100	The face attribute in the picture is illegal
4101	Face recognition algorithm is not initialized
4102	Failed to create face group
4103	Algorithm eigenvalue versions do not match
4104	The stored face bank group does not exist
4105	The number of face banks reaches the upper limit
4106	The face pixels in the picture are too large. The maximum face pixels in the picture cannot exceed 300x300
4107	The face in the picture is too blurred
4108	The angle of the face in the picture is too large
4109	Face pixels in the picture are too small
4110	Face detection failed, please try another picture and try again
4111	Person picture already exists
4196	Face library name already exists
4197	Person's name already exists
4198	The total number of face banks reached the upper limit
4199	The total number of personnel reaches the upper limit
4200	Person uniqueId already exists
4201	Personid and uniqueId are invalid
4202	Personid and uniqueId do not match
4203	FaceBank Group GroupId is invalid
4204	Invalid personid
4205	Person uniqueid is invalid

NOTE：For the upgrade, when the return value is greater than zero, indicating that the upgrade fails.

9 Services and General Resources

9.1 /Network

/Network	General Resource v1.0
GET	Viewer
Description	It is used to get the device network configuration
Query	None
Inbound Data	None
Success Return	**Network**
PUT	Viewer
Description	It is used to set the device network configuration
Query	None
Inbound Data	**Network**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

Network XML Block

```
<Network    version = "1.0" xmlns = "">
    <PPPOE>    <!-- opt -->
    <DDNS>    <!-- opt-->
    <UPNP>    <!-- opt -->
    <EMail>    <!-- opt-->
    <Port>    <!-- opt-->
    <P2P>    <!-- opt-->
    <Interfaces><!--opt-->
</Network>
```

9.1.1 /Network/PPPOE

/Network/PPPOE	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device PPPOE configuration
Query	None
Inbound Data	None
Success Return	**PPPOE**
PUT	Viewer
Description	It is used to set all of the device PPPOE configuration
Query	None

Inbound Data	**PPPOE**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

PPPOE XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<PPPOE>
    <Config/> <!-- req -->
    <Status /><!—opt-->
</PPPOE>
```

9.1.1.1 /Network/PPPOE/Config

/Network/PPPOE/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the readable and writable configuration of the device PPPOE
Query	None
Inbound Data	None
Success Return	**Config**
PUT	Viewer
Description	It is used to set the readable and writable configuration of the device PPPOE
Query	None
Inbound Data	**Config**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。 When it is retrieved the <Config> XML, <Password> field is not displayed ,but if there is not <Password> in the configuration settings, the password is not change, Otherwise, the password will be changed.	

PPPOE configuration XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
    <Enable><!--req,xs:boolean--></Enable>
    <UserName><!--req,xs:string--></UserName>
    <Password><!--wo,opt,xs:string--></Password>
</Config>
```

9.1.1.2 /Network/PPPOE/Status

/Network/PPPOE/Status	General Resource v1.0
GET	Viewer
Description	It is used to get the read-only configuration of the device PPPOE
Query	None
Inbound Data	None
Success Return	**Status**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

When <State> is the value of “connected” , <IPAddress>, <SubnetMask> and <Gateway> will be valid..

PPPOE Status XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Status>
    <State><!--ro,req,xs:string,connecting,connected,disconnected,disconnecting--></State>
    <IPAddress><!--ro,req,xs:string--></IPAddress>
    <SubnetMask>
        <!-- ro,req, xs:string, subnet mask for IPv4 address -->
    </SubnetMask>
    <Gateway><!--ro,req,xs:string--></Gateway>
</Status>
```

9.1.2 /Network/DDNS

/Network/DDNS	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device DDNS configuration

Query	None
Inbound Data	None
Success Return	**DDNS**
PUT	Viewer
Description	It is used to set all of the device DDNS configuration
Query	None
Inbound Data	**DDNS**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

DDNS XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DDNS>
    <ServerList/><!-- opt -->
    <Config/><!--req-->
    <Status/><!--opt-->
</DDNS>
```

9.1.2.1 /Network/DDNS/ServerList

/Network/DDNS/ServerList	General Resource v1.0
GET	Viewer
Description	It is used to get the read-only Server List of the device DDNS
Query	None
Inbound Data	None
Success Return	**ServerList**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

ServerList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
< ServerList>
    <Server><!--opt-->
        <ServerName><!--req,xs:string--></ServerName>
    </Server>
```

</ ServerList>

9.1.2.2 /Network/DDNS/Config

/Network/DDNS/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the readable and writable configuration of the device DDNS
Query	None
Inbound Data	None
Success Return	**Config**
PUT	Viewer
Description	It is used to Set the readable and writable configuration of the device DDNS
Query	None
Inbound Data	**Config**
Success Return	**ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

When it is retrieved the <Config> XML, <Password> field is not displayed ,but if there is not <Password> in the configuration settings, the password is not change, Otherwise, the password will be changed.

When setting <ProtocolName> field, we must ensure that the value of the field is obtained in the protocol list, otherwise you will set up unsuccessful,and no error message.

Only in <Enable> is true, the other is set to be effective.

<ServerPort> field temporarily does not support setting, the default value is 0.

DDNS Confg XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
    <Enable><!--req,xs:boolean--></Enable>
    <ProtocolName>
        <!--req,xs:string,”the string is the one of <ServerList>”>
    </ProtocolName>
    <ServerPort><!--req,xs:integer--></ServerPort>
    <DomainName><!--req,xs:string--></DomainName>
    <UserName><!--req,xs:string--></UserName>
    <Password><!--wo,req,xs:string--></Password>
    <RefreshIntervals><!--req,xs:integer, in sec--></RefreshIntervals>
</Config>
```

9.1.2.3 /Network/DDNS/Status

/Network/DDNS/ Status	General Resource v1.0
GET	Viewer
Description	It is used to get the read-only configuration of the device DDNS
Query	None
Inbound Data	None
Success Return	**Status**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

DDNS Status XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Status>
    <State>
        <!--ro,req,xs:string, “stopped,Registering ,Registered ,stopping”-->
    </State>
</Status>
```

9.1.3 /Network/UPNP

/Network/UPNP	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device UPNP configuration
Query	None
Inbound Data	None
Success Return	**UPNP**
PUT	Viewer
Description	It is used to set all of the device UPNP configuration
Query	None
Inbound Data	**UPNP**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the	

network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

UPNP XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<UPNP>
    <Config/><!--req-->
    <Status/><!--opt-->
</UPNP>
```

9.1.3.1 /Network/UPNP/Config

/Network/UPNP/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the readable and writable configuration of the device UPNP
Query	None
Inbound Data	None
Success Return	**Config**
PUT	Viewer
Description	It is used to set the readable and writable configuration of the device UPNP
Query	None
Inbound Data	**Config**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

UPNP Config XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
  <Enable><!--req,xs:boolean--></Enable>
  <WanIPAddress><!--req,xs:string--></WanIPAddress>
  <LocalHTTPPort><!--req,xs:integer--></LocalHTTPPort>
  <ExportHTTPPort><!--req,xs:integer--></ExportHTTPPort>
  <LocalMobliePort><!--req,xs:integer--></LocalMobliePort>
  <ExportMobliePort><!--req,xs:integer--></ExportMobliePort>
  <LocalCommandPort><!--req,xs:integer--></LocalCommandPort>
  <ExportCommandPort><!--req,xs:integer--></ExportCommandPort>
  <LocalMediaPort><!--req,xs:integer--></LocalMediaPort>
```

```
  <ExportMediaPort><!--req,xs:integer--></ExportMediaPort>
</Config>
```

9.1.3.2 /Network/UPNP/Status

/Network/UPNP/Status	General Resource v1.0
GET	Viewer
Description	It is used to get the read-only configuration of the device UPNP
Query	None
Inbound Data	None
Success Return	**Status**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

UPNP Status XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Status>
    <HTTPPortState><!--ro,req,xs:string,”success,failed,mapping” --></HTTPPortState>
    <MobliePortState><!--ro,req,xs:string,”success,failed,mapping” --></HTTPMoblieState>
    <CommandPortState><!--ro,req,xs:string,”success,failed,mapping”--></CommandPortState>
    <MediaPortState><!--ro,req,xs:string,”success,failed,mapping”--></MediaPortState>
</Status>
```

9.1.4 /Network/Email

/Network/Email	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device Email configuration
Query	None
Inbound Data	None
Success Return	**EMail**
PUT	Viewer
Description	It is used to set all of the device Email configuration
Query	None
Inbound Data	**EMail**

Success Return	ResponseStatus

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

If the value of <AllDay> is true , indicates all time every day is valid for the motion,but the value of <TimeBlockList> is invalid,if it is false,the value of <TimeBlockList> is valid.

A day is divided into 48 block,which means that every half hour to a block of time,when a block of time is armed,then the block is set to 1,otherwise,the block is set to 0,for example, if all time of a day is valid,then the value of <TimeBlock_0> is "111111111111111111111111111111111111111111111111",if all time of a day is invalid,then the value of <Block_0> is "000000000000000000000000000000000000000000000000".

For the <Schedule> field of the Email ,some device is not support ,some devices is support，the <Schedule> field is invalid for the not support device.

When it is retrieved the <EMail> XML, <SenderEmailPassword> field is not displayed ,but if there is not <Password> in the configuration settings, the password is not change, Otherwise, the password will be changed.

EMail XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<EMail>
    <Enable><!--req,xs:boolean--></Enable>
    <SMTPServer><!--req,xs:string--></SMTPServer>
    <SMTPPort><!--req,xs:integer--></SMTPPort>
    <SMTPAuthMethod><!--req,xs:string,None,Login--></SMTPAuthMethod>
    <SMTPTLSConfig><!--req-->
        <SMTPTLS><!--req,xs:string,"off(NONE),on(SSL),TLS,STARTTLS"--></SMTPTLS>
    </SMTPTLSConfig>
    <SenderEmailAddress><!--req,xs:string--></ SenderEmailAddress>
    <SenderEmailPassword><!--opt,xs:string--></ SenderEmailPassword>
    <ReceiverList><!--req-->
        <Receiver><!--opt-->
            <ID><!--req,xs:integer--></ID>
            <EmailAddress><!--req,xs:string--></EmailAddress>
        </Receiver>
    </ReceiverList>
    <AttachedSnapshot><!--req,xs:boolean--></AttachedSnapshot>
    <RefreshIntervals><!--req,xs:integer,in sec--></RefreshIntervals>
    <Topic><!--opt,xs:string--></Topic>
    <Message><!--opt,xs:string--></Message>
    <Schedule><!--opt-->
        <AllDay><req,xs:boolean></AllDay>
        <TimeBlockList>
            <TimeBlock_0><!req,xs:string></TimeBlock_0>
```

```
                <TimeBlock_1><!req,xs:string></TimeBlock_1>
                <TimeBlock_2><!req,xs:string></TimeBlock_2>
                <TimeBlock_3><!req,xs:string></TimeBlock_3>
                <TimeBlock_4><!req,xs:string></TimeBlock_4>
                <TimeBlock_5><!req,xs:string></TimeBlock_5>
                <TimeBlock_6><!req,xs:string></TimeBlock_6>
            <TimeBlockList>
        </Schedule>
</EMail>
```

9.1.5 /Network/Port

/Network/Port	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device Port configuration
Query	None
Inbound Data	None
Success Return	**Port**
PUT	Viewer
Description	It is used to set all of the device Port configuration
Query	None
Inbound Data	**Port**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。 If you set the this configuration , the device will be restart.	

Port XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Port>
    <HTTP><!--req,xs:integer--></HTTPPort>
    <Command><!--req,xs:integer--></CommandPort>
    <Media><!--req,xs:integer--></MediaPort>
    <Mobile><!--req,xs:integer--></MobilePort>
    <RTP><!--req,xs:integer--></RtpPort>
    <Discovery><!--req,xs:integer--></Discovery>
</Port>
```

9.1.6 /Network/P2P

/Network/P2P	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device P2P configuration
Query	None
Inbound Data	None
Success Return	**P2P**
PUT	Viewer
Description	It is used to set all of the device P2P configuration
Query	None
Inbound Data	**P2P**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。 After setting the UUID ,IPC will automatically restart, UID will take effect after the restart.	

P2P XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<P2P>
    <Enable><!--req,xs:boolean--></Enable>
    <UUID><!--req,xs:string--></UUID>
    <Status><!--ro,opt,xs:string,"offline,internet-ready,intranet-ready" --></Status>
</P2P>
```

9.1.7 /Network/P2PV2

/Network/P2PV2	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device P2P List configuration
Query	None
Inbound Data	None
Success Return	**P2PList**
PUT	Viewer
Description	It is used to set all of the device P2P configuration
Query	None

Inbound Data	P2P
Success Return	ResponseStatus
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。 After setting the UUID ,IPC will automatically restart, UID will take effect after the restart.	

P2P XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<P2PList>
    <ID><!--req,xs:integer--></ID>
    <Enable><!--req,xs:boolean--></Enable>
    <P2PProtocolName><!--req,xs:string--></P2PProtocolName>
    <UUID><!--req,xs:string--></UUID>
    <AndroidAppDownload><!--req,xs:string--></AndroidAppDownload>
    <IphoneAppDownload><!--req,xs:string--></IphoneAppDownload>
    <Status><!--ro,opt,xs:string,”offline,internet-ready,intranet-ready”--></Status>
</P2PList>
```

9.1.8 /Network/FTP

/Network/FTP	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device FTP configuration
Query	None
Inbound Data	None
Success Return	FTP
PUT	Viewer
Description	It is used to set all of the device FTP configuration
Query	None
Inbound Data	FTP
Success Return	ResponseStatus
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。 <TransferContentTypeMask> is a mask,like “00”,every bit represent a option,if the value of the bit is “0”,It is represent that it is selected,otherwise,not selected,default,From right to left, the first bit is “Image”,the second bit is “video”.	

When it is retrieved the <FTP> XML, <Password> field is not displayed ,but if there is not <Password> in the configuration settings, the password is not change, Otherwise, the password will be changed.

FTP XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<FTP Version="1.0" xmlns="http://www.zwcloud.wang/ver10/XMLSchema">
    <Enable><!—req,xs:boolean--></Enable>
    <UserName><!—req,xs:string--></UserName>
    <Password><!--wo,opt,xs:string--></Password>
    <IPAddress><!—req,xs:string--></IPAddress>
    <Port><!—req,xs:integer--></Port>
    <TransferContentTypeMask><!—req,xs:string--></TransferContentTypeMask>
    <FTPMode><!—req,xs:string,"Active, Passive"--></FTPMode>
    <StorageDirectory><!—req,xs:string--></StorageDirectory>
    <Status><!—opt,xs:string,"not connect","connected"--></Status>
</FTP>
```

9.1.9 /Network/WifiParamTest

/Network/WifiParamTest	General Resource v1.0
PUT	**Viewer**
Description	It is used to set all of the device ConnectionConfig configuration
Query	None
Inbound Data	**ConnectionConfig**
Success Return	**ResponseStatus**
Notes:	

ConnectionConfig XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<ConnectionConfig> <!--opt-->
    <SSID><!--req,xs:string--></SSID>
    <MACAddress><!--ro,req,xs:string--></MACAddress>
    <NetworkType><!--req,xs:string,managed,Ad-hoc--></NetworkType>
    <AuthMode><!--req,xs:string,OPEN,SHRED,WEPAUTO,WPAPSK,WPA2PSK,
    WPANONE,WPA,WPA2,NOPASSWORD--></AuthMode>
    <WEP><!--dep,depends on <AuthMode>-->
        <KeyID><!--req,xs:integer,"1-4"--></KeyID>
        <keyList>
            <Key>
```

```
                <ID><!—req,xs:integer--></ID>
                <KeyType><!—req,xs:string,”HEX,ASCII”--></KeyType>
                <KeyValue><!--opt,xs:string--></KeyValue>
            </Key>
        </keyList>
    </WEP>
    <WPA><!--dep,depends on <AuthMode>-->
        <EncryptType><!--req,xs:string,NONE,WEP,TKIP,AES--></EncryptType>
        <Key>
            <KeyType><!--req,xs:string,”ASCII”--></KeyType>
            <KeyValueBase64><!--req,xs:string,”the length of the source string is 8-31, and this
value will be codec by base64”--></KeyValueBase64>
        </key>
    </WPA>
</ConnectionConfig>
```

9.1.10 /Network/WifiParamTestStatus

/Network/WifiParamTestStatus	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device WIFIStatusForTest configuration
Query	None
Inbound Data	None
Success Return	**WIFIStatusForTest**
Notes:	

WIFIStatusForTest XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<WIFIStatusForTest Version="1.0">
    <Status><!--req,xs:string,”disconnect,connected,passwd error”--></Status>
</WIFIStatusForTest>
```

9.1.11 /Network/WhichNetworkCard

/Network/WhichNetworkCard	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device WhichNetcard

	configuration
Query	None
Inbound Data	None
Success Return	**WhichNetcard**
Notes:	

WhichNetcard XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<WhichNetcard Version="1.0">
    <NetcardType><!—xs:string,”Unknown,wired1,wired2,WIFI,3G,4G,5G”--></NetcardType>
</WhichNetcard>
```

9.1.12 /Network/Interfaces

/Network/Interfaces	General Resource v1.0
GET	Viewer
Description	It is used to get all of the device Interfaces configuration
Query	None
Inbound Data	None
Success Return	**InterfaceList**
PUT	Viewer
Description	It is used to set all of the device Interfaces configuration
Query	None
Inbound Data	**InterfaceList**
Success Return	**ResponseStatus**
Notes: Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。	

Network interface XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<InterfaceList>
    <Interface><!--opt-->
</InterfaceList>
```

9.1.12.1 /Network/Interfaces/NetworkCardID

/Network/Interfaces/NetworkCardID	General Resource v1.0
GET	Viewer
Description	It is used to get one of the device Interfaces configuration by the NetworkCardID.
Query	None
Inbound Data	None
Success Return	
PUT	Viewer
Description	It is used to set one of the device Interfaces configuration by the NetworkCardID.
Query	None
Inbound Data	
Success Return	**ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

By default, the value of "NetworkCardID" is 1 which represents a wired network, the value of "NetworkCardID" is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

Wire XML BLock

```
<?xml version="1.0" encoding="utf-8"?>
<Interface>
    <ID><!--req,xs:integer--></ID>
    <NetworkCardMode><!--req,xs:string--></NetworkCardMode>
    <IPAddress/><!--req-->
</Interface>
```

9.1.12.1.1 /Network/Interfaces/NetworkCardID/IPAddress

/Network/Interfaces/NetworkCardID/IPAddress	General Resource v1.0
GET	Viewer
Description	It is used to get the ip address of a particular network interface
Query	None
Inbound Data	None
Success Return	**IPAddress**

PUT	Viewer
Description	It is used to set the ip address of a particular network interface
Query	None
Inbound Data	**IPAddress**
Success Return	**ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

By default, the value of "NetworkCardID" is 1 which represents a wired network, the value of "NetworkCardID" is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

IPAddress XML BLock

```
<?xml version="1.0" encoding="utf-8"?>
<IPAddress>
    <ipVersion><!--ro,opt,xs:string,V4--></ipVersion>
        <ipAddress><!--req,xs:string--></ipAddress>
    <MacAddress><!--ro,opt,xs:string,readonly--></MacAddress>
    <SubnetMask>
        <!-- req, xs:string, subnet mask for IPv4 address -->
     </SubnetMask>
    <Gateway><!--req,xs:string--></ Gateway>
    <DHCP/><!--req-->
    <DNS/><!--req-->
</IPAddress>
```

9.1.12.1.1.1 /Network/Interfaces/NetworkCardID/IPAddress/DHCP

/Network/Interfaces/NetworkCardID/IPAddress/DHCP	General Resource v1.0
GET	Viewer
Description	It is used to get the DHCP of a particular network interface.
Query	None
Inbound Data	None
Success Return	**DHCP**
PUT	Viewer
Description	It is used to set the DHCP of a particular network interface.
Query	None
Inbound Data	**DHCP**
Success Return	**ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

Network Card 1 is support the URI /Network/Interfaces/NetworkCardID/IPAddress/DHCP,but Network Card 2 is not support..

The value of the <Status> is invalid in the Network Card 2.

By default, the value of "NetworkCardID" is 1 which represents a wired network, the value of "NetworkCardID" is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

DHCP XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DHCP>
    <Enable><!--req,xs:boolean--></Enable>
    <Status><!--ro,opt,xs:string,"initializing,obtaining,obtained" --></Status>
</DHCP>
```

9.1.12.1.1.2 /Network/Interfaces/NetworkCardID/IPAddress/DNS

/Network/Interfaces/NetworkCardID/IPAddress/DNS	General Resource v1.0
GET	Viewer
Description	It is used to get the DNS of a particular network interface.
Query	None
Inbound Data	None
Success Return	**DNS**
PUT	Viewer
Description	It is used to set the DNS of a particular network interface.
Query	None
Inbound Data	**DNS**
Success Return	ResponseStaus **ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

Network Card 1 is support the URI /Network/Interfaces/NetworkCardID/IPAddress/DNS,but Network Card 2 is not support.

The value of The <Enable> in the Network Card 2 is depended on the value of the <Enable> in the <DHCP> ,so It is invalid that you set the value of the <Enable> in the <DNS>.

By default, the value of "NetworkCardID" is 1 which represents a wired network, the value of "NetworkCardID" is 2 which represents a wireless network card.For the future, to support

more network card, then turn increases here does not support more than 2.

DNS XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DNS>
    <Enable><!--req,xs:boolean--></Enable>
    <PrimaryDNS><!--req,xs:string--></PrimaryDNS>
    <SecondaryDNS><!--req,xs:string--></SecondaryDNS>
</DNS>
```

9.1.12.1.2 /Network/Interfaces/ NetworkCardID/WIFIAccessPointList

/Network/Interfaces/NetworkCardID/WIFIAccessPointList	General Resource v1.0
GET	Viewer
Description	It is used to get all wireless Access Point List.
Query	None
Inbound Data	None
Success Return	**WIFIAccessPointList**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

Only wireless card supports the URI /Network/Interfaces/NetworkCardID/WIFIAccessPointList.

WirelessMode:

1. legacy 11B only
2. legacy 11A only
3. legacy 11a/b/g mixed
4. legacy 11G only
5. 11ABGN mixed
6. 11N only
7. 11GN mixed
8. 11AN mixed
9. 11BGN mixed

By default, the value of “NetworkCardID” is 1 which represents a wired network, the value of “NetworkCardID” is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

WIFIAccessPointList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<WIFIAccessPointList><!--req-->
  <AccessPoint><!--opt-->
    <ID><!--ro,req,xs:integer--></ID>
```

```
        <SSID><!ro,req,xs:string--></SSID>
        <MACAddress><!--ro,req,xs:string--></MACAddress>
        <SignalValue><!--ro,req,xs:integer,0-100--></SignalValue>
        <Authentication><!--ro,req,xs:string,None,shared,WPAPSK,WPA2PSK,
        WPANONE,WPA,WPA2--></Authentication>
        <EncryptType><!--ro,req,xs:string,NONE,WEP,TKIP,AES--></EncryptType>
        <WirelessMode><!--ro,req,xs:integer--></WirelessMode>
      </AccessPoint>
</WIFIAccessPointList>
```

9.1.12.1.3 /Network/Interfaces/NetworkCardID/Wireless

/Network/Interfaces/NetworkCardID/Wireless	General Resource v1.0
GET	Viewer
Description	It is used to get the WIFI information of a wireless network interface.
Query	None
Inbound Data	None
Success Return	**ConnectionConfig**
PUT	Viewer
Description	It is used to set the WIFI information of a wireless network interface.
Query	None
Inbound Data	**ConnectionConfig**
Success Return	**ResponseStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

Only wireless card supports the URI /Network/Interfaces/NetworkCardID/WIFIAccessPointList.

If <AuthMode> mode is OPEN, SHARED mode, then the <WEP> is Effective. If <AuthMode> model is WPAPSK, WPA2PSK more, then the <WPA> is Effective. Otherwise, they are all not valid.but they are all displayed.

if the <AuthMode> is WPAPSK or WPA2PSK,the value of the <keyvalue> in the <WPA> should not be empty.

The value of the <KeyType> in the <WPA> must be “ASCII”.

By default, the value of “NetworkCardID” is 1 which represents a wired network, the value of “NetworkCardID” is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

WIFI ConnectionConfig XML Block

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ConnectionConfig> <!--opt-->
    <Enable><!--req,xs:integer,"disconnected,connected" --></Enable>
    <SSID><!--req,xs:string--></SSID>
    <MACAddress><!--ro,req,xs:string--></MACAddress>
    <NetworkType><!--req,xs:string,managed,Ad-hoc--></NetworkType>
    <AuthMode><!--req,xs:string,OPEN,SHRED,WEPAUTO,WPAPSK,WPA2PSK,
    WPANONE,WPA,WPA2,NOPASSWORD--></AuthMode>
    <WEP><!--dep,depends on <AuthMode>-->
        <KeyID><!--req,xs:integer,"1-4"--></KeyID>
        <keyList>
            <Key>
                <ID><!—req,xs:integer--></ID>
                <KeyType><!—req,xs:string,"HEX,ASCII"--></KeyType>
                <KeyValue><!--opt,xs:string--></KeyValue>
            </Key>
        </keyList>
    </WEP>
    <WPA><!--dep,depends on <AuthMode>-->
        <EncryptType><!--req,xs:string,NONE,WEP,TKIP,AES--></EncryptType>
        <Key>
            <KeyType><!--req,xs:string,"ASCII"--></KeyType>
            <KeyValueBase64><!--req,xs:string,"the length of the source string is 8-31, and this
value will be codec by base64"--></KeyValueBase64>
        </key>
    </WPA>
</ConnectionConfig>
```

9.1.12.1.4 /Network/Interfaces/NetworkCardID/WIFIStatus

/Network/Interfaces/NetworkCardID/WIFIStatus	General Resource v1.0
GET	Viewer
Description	It is used to get the status of a wireless network interface.
Query	None
Inbound Data	None
Success Return	**WIFIStatus**

Notes:

Please do not frequently set the network configuration, because the equipment end of the set the network configuration is asynchronous responses, so if you frequently set, only the response to the first operation, the recommended time interval set 3-5s。

Only wireless card supports the URI /Network/Interfaces/NetworkCardID/WIFIAccessPointList.

By default, the value of "NetworkCardID" is 1 which represents a wired network, the value

of “NetworkCardID” is 2 which represents a wireless network card.For the future, to support more network card, then turn increases here does not support more than 2.

WIFIStatus XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<WIFIStatus><!--req-->
    <State><!--ro,req,xs:string,”open,close,notsupport” --></State>
</WIFIStatus>
```

9.1.12.2 /Network/Interfaces/3/4GCardCfg

Network/Interfaces/3/4GCardCfg	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of 3/4G card
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or _4GCardCfg
PUT	Operator
Description	It is used to set the configuration of 3/4G card
Query	None
Inbound Data	**_4GCardCfg**
Success Return	**ResponseStatus**
Notes: Not all devices support.	

_4GCardCfg XML

```
<?xml version="1.0" encoding="utf-8"?>
<_4GCardCfg>
    <APNMode><!—rw,req,xs:string--></APNMode>
    <APNServer><!—rw,req,xs:string--></APNServer>
    <APNUserName><!—rw,req,xs:string--></APNUserName>
    <APNPassword><!—rw,req,xs:string--></APNPassword>
    <RegisterStatus><!—rw,req,xs:Interger--></RegisterStatus>
    <SignalVal><!—rw,req,xs:string--></SignalVal>
    <EnableModWander><!—rw,req,xs:string--></EnableModWander>
    <IPAddress><!—rw,req,xs:string--></IPAddress>
    <Gateway><!—rw,req,xs:string--></Gateway>
</_4GCardCfg>
```

9.1.12.3 /Network/Interfaces/3/4GCardInfo

Network/Interfaces/3/4GCardInfo	General Resource v1.0

GET	Viewer
Description	It is used to get the info of 3/4G card
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or _4GCardInfo
Notes: Not all devices support.	

_4GCardCfg XML

```
<?xml version="1.0" encoding="utf-8"?>
<_4GCardInfo>
    <IMSI><!—ro,req,xs:string--></IMSI>
    <IMEI><!—ro,req,xs:string--></IMEI>
    <ACT><!—ro,req,xs:string--></ACT>
    <Operator><!—ro,req,xs:string--></Operator>
    <Band><!—ro,req,xs:string--></Band>
    <ICCID><!—ro,req,xs:string--></ICCID>
</_4GCardInfo>
```

9.1.12.4 /Network/Interfaces/3/4GCardTest

Network/Interfaces/3/4GCardTest	General Resource v1.0
PUT	Operator
Description	It is used to test 3/4G card
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: Not all devices support.	

9.1.13 /Network/CloudStorage

Network/CloudStorage	General Resource v1.0
GET	Viewer
Description	It is used to get the device ID of CloudStorage
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **CloudStorage**
PUT	Operator
Description	It is used to set the device ID of CloudStorage
Query	None
Inbound Data	**CloudStorage**

Success Return	ResponseStatus
Note:	

CloudStorage XML

```
<?xml version="1.0" encoding="utf-8"?>
<CloudStorage>
    <DeviceID><!—ro,req,xs:string--></DeviceID>
</CloudStorage>
```

9.1.13.1 /Network/CloudStroage/BaseConfig

/Network/CloudStorage/BaseConfig	General Resource v1.0
GET	Viewer
Description	It is used to get the base config of cloud storage.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or BaseConfig**
PUT	**Operator**
Description	It is used to set the base config of cloud storage.
Query	None
Inbound Data	**BaseConfig**
Success Return	ResponseStaus
Notes:	

9.1.13.2 /Network/CloudStroage/OpenStatus

/Network/CloudStorage/OpenStatus	General Resource v1.0
GET	Viewer
Description	It is used to get the open status of cloud storage.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or OpenStatus**
PUT	Operator
Description	It is used to set the open status of cloud storage.
Query	None
Inbound Data	**OpenStatus**

Success Return	ResponseStaus
Notes:	

OpenStatus XML

```
<?xml version="1.0" encoding="utf-8"?>
<OpenStatus>
    <SubscribeStatus><!—ro,req,xs:string”true”,”false”--></SubscribeStatus>
    <DealID><!—ro,req,xs:string--></DealID>
</OpenStatus>
```

9.1.13.3 /Network/CloudStroage/RecordPlan

/Network/CloudStorage/RecordPlan	General Resource v1.0
GET	Viewer
Description	It is used to get the record plan of cloud storage.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or RecordPlan**
PUT	Operator
Description	It is used to set the record plan of cloud storage.
Query	None
Inbound Data	**RecordPlan**
Success Return	ResponseStaus
Notes:	

RecordPlan XML

```
<?xml version="1.0" encoding="utf-8"?>
<RecordPlan>
    <StreamMode><!—ro,req,xs:string”MainStream”,”SubStream”,”SecondSubStream”--></Stre
amMode>
    <RecordMode><!—ro,req,xs:string”ManualRecord”,”AutoRecord”,”CloseRecord”--></Record
Mode>
    <TimeBlockList>
    <TimeBlock_0>111111111111111111111111111111111111111111111111</TimeBlock_0>
    <TimeBlock_1>111111111111111111111111111111111111111111111111</TimeBlock_1>
    <TimeBlock_2>111111111111111111111111111111111111111111111111</TimeBlock_2>
    <TimeBlock_3>111111111111111111111111111111111111111111111111</TimeBlock_3>
    <TimeBlock_4>111111111111111111111111111111111111111111111111</TimeBlock_4>
    <TimeBlock_5>111111111111111111111111111111111111111111111111</TimeBlock_5>
    <TimeBlock_6>111111111111111111111111111111111111111111111111</TimeBlock_6>
    </TimeBlockList>
</RecordPlan>
```

9.1.13.4 /Network/CloudStroage/BaseStatus

/Network/CloudStorage/BaseStatus	General Resource v1.0
GET	Viewer
Description	It is used to get the base status of cloud storage.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or BaseStatus**
PUT	Operator
Description	It is used to set the base status of cloud storage.
Query	None
Inbound Data	**BaseStatus**
Success Return	ResponseStaus
Notes:	

BaseStatus XML

```
<?xml version="1.0" encoding="utf-8"?>
<BaseStatus>
    <NetworkStatus><!—ro,req,xs:Interger--></NetworkStatus>
    <StorageStatus><!—ro,req,xs:Interger--></StorageStatus>
    <ErrorCode><!—ro,req,xs:Interger--></ErrorCode>
</BaseStatus>
```

9.1.13.5 /Network/CloudStroage/BusinessInfo

/Network/CloudStorage/BusinessInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the business info of cloud storage.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or BusinessInfo**
PUT	Operator
Description	It is used to set the business info of cloud storage.
Query	None
Inbound Data	**BusinessInfo**
Success Return	ResponseStaus
Notes:	

BusinessInfo XML

```
<?xml version="1.0" encoding="utf-8"?>
<BusinessInfo>
    <OrderInfo>
        <Status><!--req,xs:string”unBind”,”effective”,”expired”--></Status>
        <OrderID><!--req,xs:string--></OrderID>
    </OrderInfo>
    <PackageInfo>
        <PackageName><!—req,xs:string--></PackageName>
        <PackageBeginTime><!—req,xs:datetime”20200220T000000”--></PackageBeginTime>
        <PackageEndTime><!—req,xs:datetime”20200220T000000”--></PackageEndTime>
        <CycleDays><!—req,xs:Interger--></CycleDays>
        <RemainDays><!—req,xs:Interger--></RemainDays>
    </PackageInfo>
</BusinessInfo>
```

9.2 /System

/System	General Resource v1.0

9.2.1 /System/Time

/System/Time	General Resource v1.0
GET	Viewer
Description	It is used to get the device time information.
Query	None
Inbound Data	None
Success Return	**Time**
PUT	Viewer
Description	It is used to udpate the device time information.
Query	None
Inbound Data	**Time**
Success Return	ResponseStaus **ResponseStatus**
Notes: SystemTime:An ISO 8601 Date/Time string is accepted and returned.	

Time XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Time>
    <DateTimeFormat>
        <!--req,string,YYYYMMDDWhhmmss,YYYYMMDDhhmmss,MMDDYYYYWhhmmss,MMD
```

```
            DYYYYhhmmss,DDMMYYYYWhhmmss,DDMMYYYYhhmmss-->
    </DateTimeFormat>
    <TimeFormat><!--req,xs:string,12hour,24hour--></TimeFormat>
    <SystemTime><!--req,xs:datetime," 20040503T173008+08"--></SystemTime>
</Time>
```

9.2.2 /System/NTP

/System/NTP	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of NTP server for the device.
Query	None
Inbound Data	None
Success Return	**NTP**
PUT	Viewer
Description	It is used to set the configuration of NTP server for the device.
Query	None
Inbound Data	**NTP**
Success Return	**ResponseStatus**
Notes: TimeZone: It must be start with "CST".	

NTP XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<NTP>
    <Enable><!—req,xs:boolean--></Enable>
    <ServerName><!—req,xs:string,"example: time.windows.com"--></ServerName>
    <TimeZone><!--req, xs:string, POSIX time zone string,"CST-09:00:00"--></TimeZone>
    <TimeSyncInterval><!—req,xs:integer--></TimeSyncInterval>
</NTP>
```

9.2.3 /System/DST

/System/DST	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of DST for the device.
Query	None

Inbound Data	None
Success Return	**DST**
PUT	Viewer
Description	It is used to set the configuration of DST for the device.
Query	None
Inbound Data	**DST**
Success Return	**ResponseStatus**

Notes:

When the <Mode> is “date” ,the <DateModeStart> and<DateModeStop> is effective , When the <Mode> is “week” ,the < WeekModeStart > and< WeekModeStop > is effective

<TimeDifference> the unit is seconds,example:3600 is 3600 seconds

<DateModeStart> and <DateModeStop>:An ISO 8601 Date/Time string is accepted and returned.

<WeekModeStart> and <WeekModeStop>:the first number of the “3-1-1-2-3-6” is March,the second is the first week of the month,the third is Monday of the first week,the forth is 2 hour,the fifth is 3 minute, the sixth is 6 second

NTP XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<DST Version="1.0" xmlns="http://www.zwcloud.wang/ver10/XMLSchema">
    <Enable><!—req,xs:boolean--></Enable>
    <Mode><!—req,xs:string,”date,week”--></Mode>
    <TimeDifference>req,xs:string, </TimeDifference>
    <DateModeStart><!--req,xs:datetime,” 20040503T173008+08”--></DateModeStart>
    <DateModeStop><!--req,xs:datetime,” 20040503T173008+08”--></DateModeStop>
    <WeekModeStart><!--req,xs:string,” 3-1-1-2-3-6”--></WeekModeStart>
    <WeekModeStop><!--req,xs:string,” 2-4-5-7-8-8”--></WeekModeStop>
</DST>
```

9.2.4 /System/DeviceInfo

/System/DeviceInfo	General Resource v1.0
GET	Viewer
Description	It is used to get device information.
Query	None
Inbound Data	None
Success Return	**DeviceInfo**
PUT	Viewer
Description	It is used to set the configuration of DeviceName for the device.
Query	None

Inbound Data	DeviceInfo
Success Return	**ResponseStatus**
Notes: <DeviceName> and <ActiveCode> can be set.others are not be set. <ActiveStatus> “success” means active success, or be “invalid active code” <IsSupportActiveCode>, if this field was false, the client should not show Active Code and Active Status field.	

DeviceInfo XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DeviceInfo>
    <DeviceName><!--rw,req,xs:string--></DeviceName>
    <BoardType><!--ro,req,xs:integer--></BoardType>
    <DeviceTypeName><!--ro,req,xs:string--></DeviceTypeName>
    <SerialNum><!--ro,req,xs:string--></SerialNum>
    <HardWareVersion><!--ro,req,xs:string--></HardWareVersion>
    <SoftWareVersion><!--ro,req,xs:string--></SoftWareVersion>
    <VideoInputChannels><!-ro,-req,xs:integer--></VideoChannels>
    <AudioChannels><!--ro,req,xs:integer--></AudioChannels>
    <AlarmInputChannels><!--ro,req,xs:integer--></AlarmChannel>
    <AlarmOutputChannels><!--ro,req,xs:integer--></AlarmOutputChannel>
    <MonitorCount><!--ro,req,xs:integer--></MonitorCount>
    <SpotOutChannels><!--ro,req,xs:integer--></SpotOutChannels>
    <TwoWayAudio><!--ro,req,xs:boolean--></TwoWayAudio>
    <SmartAVCList Version="1.0" xmlns="http://www.zwcloud.wang/ver10/XMLSchema">
        <SmartAVC Version="1.0" xmlns="http://www.zwcloud.wang/ver10/XMLSchema">
            <Support><!--ro,opt,xs:Boolean--></Support>
        </SmartAVC>
    </SmartAVCList>
    <IsSupportActiveCode><!--ro,req,xs:boolean--></IsSupportActiveCode>
    <ActiveCode><!--ro,req,xs:string--></ActiveCode>
    <ActiveStatus><!--ro,req,xs:string--></ActiveStatus>
    // Algorithm Version, The version that describes the ai algorithm
    <AlgorithmVersion><!--ro,req,xs:string--></AlgorithmVersion>
</DeviceInfo>
```

9.2.5 /System/DeviceStatus

/System/DeviceStatus	General Resource v1.0
GET	Viewer
Description	It is used to get all Channels Status of the device

Query	None
Inbound Data	None
Success Return	**DevicestatusList**
Notes:	

DevicestatusList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DeviceStatusList>
    <DeviceStatus/><!--opt-->
</DeviceStateList>
```

9.2.5.1 /System/DeviceStatus/ChannelID

/System/DeviceStatus/ID	General Resource v1.0
GET	Viewer
Description	It is used to get a Channel Status of the device
Query	None
Inbound Data	None
Success Return	**DeviceStatus**
Notes: The value of the <FirstSubStreamEnable> field represents whether It is open the first sub stream or not. The value of the <SecondSubStreamEnable> field represents whether It is open the second sub stream or not. The value of the <RecordState> field is “none”,represents“not support” for the Unsupported device,but “no records” for the supported device. The value of the <VideoLostState> field is “false”,represents“not support” for the unsupported device,but “no video lost event” for the supported device. The value of the < MotionState > field is “false”,represents“not support” for the unsupported device,but “no motion event” for the supported device. The value of the < AlarmInState > field is ”false”,represents“not support” for the unsupported device,but “no alarm in event” for the supported device. The value of the < AlarmInState > field is “false”,represents“not support” for the unsupported device,but “no alarm out event” for the supported device.	

DeviceStatus XML Block

```
<DeviceStatus>
    <ID><!--req,xs:integer--></ID>
    <MainStreamVideoBitrate><!--ro,req,xs:integer --></ MainStreamVideoBitrate>
    <FirstSubStreamEnable><!--ro,req,xs:boolean--></ FirstSubStreamEnable>
```

```
    <FirstSubStreamVideoBitrate><!--ro,req,xs:integer--></ FirstSubStreamVideoBitrate>
    <SecondSubStreamEnable><!--ro,req,xs:boolean--></ SecondSubStreamEnable>
    <SecondSubStreamVideoBitrate><!--ro,req,xs:integer--></SecondStreamVideoBitrate>
    <RecordState><!--ro,req,xs:string,"none,alarm,normal,manual,motion"--></RecordState>
    <VideoLostState><!--ro,req,xs:boolean--></VideoLostState>
    <MotionState><!--ro,req,xs:boolean--></MotionState>
    <AlarmInState><!--ro,req,xs:boolean--></AlarmInState>
    <AlarmOutState><!--ro,req,xs:boolean--></AlarmOutState>
</DeviceStatus>
```

9.2.6 /System/FirmwareUpgrade

/System/FirmwareUpgrade	General Resource v1.0
PUT	Viewer
Description	It is used to upgrade the firmware of the device.
Query	None
Inbound Data	**Opaque Data**
Success Return	**ResponseStatus**
Notes: The URI /System/FirmwareUpgrade shows the equipment upgrade, using HTTP PUT method, with authentication and update files to be uploaded.	

9.2.7 /System/Reboot

/System/Reboot	General Resource v1.0
PUT	Viewer
Description	It is used to reboot the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.2.8 /System/FactoryDefault

/System/FactoryDefault	General Resource v1.0
PUT	Viewer

Description	It is used to reset the configuration for the device to the factory default.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: It is not recovery the network configuration.	

9.2.9 /System/FactoryDefaultV2

/System/FactoryDefaultV2	General Resource v1.0
PUT	Viewer
Description	It is used to reset the configuration for the device to the factory default.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: It is recovery all configuration.	

9.2.10 /System/SendTestEmail

/System/SendTestEmail	General Resource v1.0
PUT	Viewer
Description	It is used to Send Test Email.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Please first properly configured Email Configuration with using the URL /Network/Email.	

9.2.11 /System/RecoveryAdminPassword

/System/RecoveryAdminPassword	General Resource v1.0
PUT	Viewer
Description	It is used to Recovery the password of the user “admin” Password And IP.

Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Defaultly the password of the user “admin” is empty,The IP is “192.168.1.11” .	

9.2.12 /System/TryListenCurrentAlarmAudio

/System/TryListenCurrentAlarmAudio	General Resource v1.0
PUT	Viewer
Description	Listen to the current alarm audio
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.2.13 /System/Logging

/System/Logging	General Resource v1.0
GET	Viewer
Description	It is used to get the log information of the device.
Query	majorType minorType startTime stopTime
Inbound Data	None
Success Return	**LogList**
Notes: The value of “majorType” is: 1:System Operation 2:Configuration Operation 3:Record Operation 4:User Managerment 5:Alarm Event 6:Exception Event 7:Storage Management 8:All Operation When the value of “majorType” is 1, the value of “minorType” is: 0: All System Operation	

1: System upgrade
2: PTZ control
3: Illegal shutdown
4: Normal shutdown
5: Startup
6: Remote reboot
7:Clear alarm
When the value of “majorType” is 2, the value of “minorType” is:
100:All Configuration Operation
101: Record time table settings
102: Normal settings
103: Encoding settings
104: Network settings
105: Server settings
106: Motion detection settings
107: PTZ decoder settings
108: Restore default settings
109: Set display device parameters
110: Display mode settings
111: Image color settings
112: Auto-maintenance settings
113: Alarm output settings
114: Alarm input settings
115: Function customize settings
116: Import PTZ protocol
117: Export PTZ protocol
118: Import parameters settings
119: Export parameters settings
120: Abnormal parameters settings
When the value of “majorType” is 3, the value of “minorType” is:
200: All Record Operation
201: Start manual record
202: Start auto record
203: Stop
When the value of “majorType” is 4, the value of “minorType” is:
300:All User Managerment
301: Add user
302: Delete user
303: Modify user
304: User login
305: User logout
When the value of “majorType” is 5, the value of “minorType” is:
400: All Alarm Event
401: Motion detection begin

402: Motion detection end
403: Input alarm begin
When the value of “majorType” is 6, the value of “minorType” is:
500: All Exception Event
501: IP confliction
502: HDD full
503: HDD error
504: Illegal visit
505: Signal loss
506: Signal restore
507: Network disconnect
When the value of “majorType” is 7, the value of “minorType” is:
600: All Storage Management
601: Format partition
602: Create partition
603: Delete partition
604: Uninstall HDD
605: HDD group management

The format of “startTime” and “stopTime” is “YYYY-MM-DDThh:mm:ss”.
Devices support up to 1000 log.

For example:
http://192.168.1.171/System/Logging?majorType=8&minorType=0&startTime=2015-07-28T00:00:00&stopTime=2015-07-29T23:59:00.
The value of the minorType is invalid for some device.

LogList XML Block

```
<LogList>
    <Log> <!-- opt -->
        <ID> <!-- req, xs:integer --> </ID>
        <time> <!--req, xs:datetime --> </time>
        <majorType> <!--req, xs:integer --> </majorType>
        <minorType> <!--req, xs:integer --> </minorType>
        <netUser> <!--req, xs:string --> </netUser>
        <hostIPAddress> <!--req, xs:string --> </hostIPAddress>
        <channel> <!--req, xs:integer --> </channel>
    </Log>
</LogList>
```

9.2.14 /System/Users

/System/Users	General Resource v1.0

GET	Viewer
Description	It is used to get the configuration of all users for the device.
Query	None
Inbound Data	None
Success Return	**UserList**
Notes:	

NTP XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<UserList>
    <User> <!-- opt -->
</UserList>
```

9.2.15 /System/Users/ID

/System/Users/ID	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of the specifies ID user for the device.
Query	None
Inbound Data	None
Success Return	**User**
PUT	**Viewer**
Description	It is used to update the specifies user for the device.
Query	None
Inbound Data	**User**
Success Return	**ResponseStatus**
POST	**Viewer**
Description	It is used to add a user for the device.
Query	None
Inbound Data	**User**
Success Return	**ResponseStatus**
Delete	**Viewer**
Description	It is used to delete the specifies user for the device.
Query	None
Inbound Data	**User**
Success Return	**ResponseStatus**

Notes:
When call the"GET" method ,The <ID> unit is effective,when call other method , The <ID> unit is no effective.you can not set the user information by <ID>,you just set user password by <UserName>. The <ID> is start at 1. Please do not delete the "Default" or "admin" user. Please do not create a "Default" or "Adminstrator" level user,you just can create a "Normal" level user. We just support the user level ,the right for the different level is your decision.

NTP XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<User>
    <ID><!--req,xs:integer--></ID>
    <UserName><!--req,xs:string--></UserName>
    <Password><!--wo,req,xs:string--></Password>
    <UserLevel><!--req,xs:string,"Default,Adminstrator,Normal"> </UserLevel>
    <IPAddress><!--ro,opt,xs:string></IPAddress>
    <MACAddress><!--ro,opt,xs:string></MACAddress>
</User>
```

9.2.16 /System/ChannelID/RemoteForceIFrame

/System/ChannelID/RemoteForceIFrame	General Resource v1.0
PUT	Viewer
Description	It is used to Remote Force I Frame.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number.The <ChannelID> is start at 1.	

9.2.17 /System/ChannelID/RemoteAlarmOutClear/ID

/System/ChannelID/RemoteAlarmOutClear/ID	General Resource v1.0
PUT	Viewer
Description	It is used to Remote Clear Alarm out.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:
The <ChannelID> in URI shows the alarm channel number. The <ChannelID> is start at 1.The <ID> in URI shows the Alarm Type, please set the ID to 0.

9.2.18 /System/ChannelID/RemoteAlarmoutControl/on

/System/ChannelID/RemoteAlarmoutControl/on	General Resource v1.0
PUT	Viewer
Description	It is used to open alarmout state
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: The ChannelID in URI shows the alarm channel number. The <ChannelID> is start at 1. It is used to set the alarmout state to open in the /System/AlarmoutState.	

9.2.19 /System/ChannelID/RemoteAlarmoutControl/off

/System/ChannelID/RemoteAlarmoutControl/off	General Resource v1.0
PUT	Viewer
Description	It is used to close alarmout state
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: The ChannelID in URI shows the alarm channel number. The <ChannelID> is start at 1. It is used to set the alarmout state to close in the /System/AlarmoutState.	

9.2.20 /System/AlarmoutState

/System/Alarmoutstate	General Resource v1.0
GET	Viewer
Description	It is used to close alarmout
Query	None
Inbound Data	None
Success Return	**AlarmoutState**
Notes: The format of <State>is similar to “11111111” or “00000000”, Each bit represents whether	

the alarm of a channel is open or not, NO.1 means trigger the alarm-out, NO.0 means not yet trigger the alarm-out, for example, the string “11111111” Represents that the device is 8 channels, from left to right, represent channel 1,channel 2,channel 3…… and so on.

AlarmoutState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmoutState><!--req-->
    <State><!--ro,req,xs:string,”00000000000000” --></State>
</ AlarmoutState>
```

9.2.21 /System/AlarmInState

/System/AlarmInstate	General Resource v1.0
GET	Viewer
Description	It is used to close alarmout
Query	None
Inbound Data	None
Success Return	**AlarmInState**
Notes: The format of <State>is similar to “11111111” or “00000000”, Each bit represents whether the alarm of a channel is open or not, NO.1 means close, NO.0 means open, for example, the string “11111111” Represents that the device is 8 channels, from left to right, represent channel 1,channel 2,channel 3…… and so on.	

AlarmInState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmInState><!--req-->
    <State><!--ro,req,xs:string,”00000000000000” --></State>
</ AlarmInState>
```

9.2.22 /System/DeviceConfigFile

/System/DeviceConfigFile	General Resource v1.0
GET	Viewer
Description	It is used to download the device configuration file
Query	None
Inbound Data	None
Success Return	**Config file**
PUT	Viewer

Description	It is used to upload the device configuration file
Query	None
Inbound Data	**Config file**
Success Return	**ResponseStatus**
Notes:	

9.2.23 /System/TransparentChannel/WriteOnce

/System/TransparentChannel/WriteOnce	General Resource v1.0
PUT	Operator
Description	It is used to send data on the transparent channel.
Query	None
Inbound Data	**TransparentChannelWriteonce**
Success Return	**ResponseStatus**

Notes:

<Data> please put the data you want to send to the string type.the format of data is "12 23 ff fe fc",the max count of number is 32.the range of value is 0--ff

The value of "majorType" is the number such as,0,1,2....., when the value is 0,the type of Serial port is RS232,when the value is 1,the Type of Serial port is RS485,when the value is 2, the type of Serial port is RS422.

Thie value of "minorType" is also number such as 0,1,2....., default the value is 0.

Default the value of the majorType is 2 and the value of the minorType is 0.others is not support .

The baudrate is depend on the URL /PTZ/ChannelID/Config.

For example:

http://192.168.1.171/System/TransparentChannel/WriteOnce?majorType=2&minorType=0

TransparentChannelWriteOnce **XML Block**

```
<?xml version="1.0" encoding="utf-8"?>
<TransparentChannelWriteonce><!--req-->
    <Data><!--ro,req,xs:string, --></Data>
</TransparentChannelWriteonce>
```

9.2.24 /System/TransparentChannel/Connect

/System/TransparentChannel/Connect	General Resource v1.0

PUT	Operator
Description	It is used to create a connnecttion on the transparent channel.
Query	None
Inbound Data	**TransparentChannelConnectReponse**
Success Return	**ResponseStatus**

Notes:

The value of “majorType” is the number such as,0,1,2……,default the value is 0.

Thie value of “minorType” is also number such as 0,1,2……,when the value is 0,the type of Serial port is RS232,when the value is 1,the Type of Serial port is RS485,when the value is 2, the type of Serial port is RS422.

This URL is not support.

For example:

http://192.168.1.171/System/TransparentChannel/Connect?majorType=2&minorType=0

TransparentChannelConnectReponse **XML Block**

```
<?xml version="1.0" encoding="utf-8"?>
<TransparentChannelConnectReponse><!--req-->
    <Index><!--ro,req,xs:string, --></Index>
</TransparentChannelConnectReponse>
```

9.2.25 /System/TransparentChannel/Write

/System/TransparentChannel/Write	General Resource v1.0
PUT	Operator
Description	It is used to write datas on the transparent channel.
Query	None
Inbound Data	**TransparentChannelWrite**
Success Return	**ResponseStatus**

Notes:

<Index> the Index need be set, It is from the response of “/System/TransparentChannel/Connect”.

<Data> please put the data you want to send to the string type.

This URL is not support.

AlarmInState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<TransparentChannelWrite><!--req-->
    <Index><!--ro,req,xs:string, --></Index>
    <Data><!--ro,req,xs:string, --></Data>
</TransparentChannelWrite>
```

9.2.26 /System/TransparentChannel/Read

/System/TransparentChannel/Read	General Resource v1.0
PUT	Operator
Description	It is used to read datas on the transparent channel.
Query	None
Inbound Data	**TransparentChannelRead**
Success Return	**ResponseStatus**
Notes: <Index> the Index need be set, It is from the response of "/System/TransparentChannel/Connect". <Data> the data is from device. This URL is not support.	

AlarmInState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<TransparentChannelRead><!--req-->
    <Index><!--ro,req,xs:string, --></Index>
    <Data><!--ro,req,xs:string, --></Data>
</TransparentChannelRead>
```

9.2.27 /System/TransparentChannel/Disconnect

/System/TransparentChannel/Disconnect	General Resource v1.0
PUT	Operator
Description	It is used to destroy the connection on the transparent channel.
Query	None
Inbound Data	**TransparentChannelDisconnect**
Success Return	**ResponseStatus**
Notes: <Index> the Index need be set, It is from the response of "/System/TransparentChannel/Connect". This URL is not support.	

AlarmInState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<TransparentChannelDisconnect><!--req-->
```

```
    <Index><!--ro,req,xs:string, --></Index>
</TransparentChannelDisconnect>
```

9.2.28 /System/DeviceAutoMaintenance

/System/DeviceAutoMaintenance	General Resource v1.0
GET	Viewer
Description	It is used to send data on the transparent channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AutoMaintenance**
PUT	Operator
Description	It is used to send data on the transparent channel.
Query	None
Inbound Data	**AutoMaintenance**
Success Return	**ResponseStatus**

Notes:

<Enable>: when <Enable> is false ,other value is invalid, otherwise,other value is valid.

<Mode>: when <Mode> is “everyweek”,<WeekDayMask> and <MaintenanceTime> is valid,when <Mode> is “everyday” or “once”,only <MaintenanceTime>is valid.

<WeekDayMask>:the format of <WeekDayMask> is “0000000”,from left to right,it is show Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday.if the bit for one day is “1”,then this day is open automatic-mantenance,otherwise,this is close automatic-mantenance.

AlarmInState XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AutoMaintenance><!--req-->
    <Enable><!--req,xs:boolean--></Enable>
    <Mode><!--req,xs:string,”everyweek,everyday,once”--></Mode>
    <WeekDayMask><!—req,xs:string,”0000000”--></WeekDayMask>
    <MaintenanceTime><!--req,xs:datetime,” 20040503T173008”-->
    </ MaintenanceTime>
</AutoMaintenance>
```

9.2.29 /System/SendTestFTP

/System/SendTestFTP	General Resource v1.0
PUT	Viewer
Description	It is used to Send Test FTP.

Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Please first properly configured Email Configuration with using the URL /Network/FTP.	

9.2.30 /System/DeviceWifiSTA2AP

/System/DeviceWifiSTA2AP	General Resource v1.0
PUT	Viewer
Description	It is used to Enter AP Mode
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.2.31 /System/DeviceWifiAP2STA

/System/DeviceWifiSTA2AP	General Resource v1.0
PUT	Viewer
Description	It is used to Enter STA Mode
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.2.32 /System/DeviceCustomFunction

/System/DeviceCustomFunction	General Resource v1.0
PUT	Viewer
Description	It is used to custom function.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: You can transfer data that you want to transfer.	

9.2.33 /System/CloudUpgradeServerInfo

/System/CloudUpgradeServerInfo	General Resource v1.0
PUT	Viewer
Description	It is used to Set cloud upgrade server information.
Query	None
Inbound Data	**CloundUpgradeServer**
Success Return	**ResponseStatus**
Notes: The following configuration is the configuration of the upgrade server, including the IP,<HostIP>, the port number of the service,<Port>, the username,<Username> and password,<Password>, the protocol,<Protocol> used, the corresponding path,<Path>	

CloundUpgradeServer XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<CloundUpgradeServer><!--req-->
    <Port><!--req,xs:interger,0-65535--></Port>
    <HostIP><!--req,xs:string,it must be a ip address--></HostIP>
    <Username><!—req,xs:string,the max length is 256--></Username>
    <Password><!--req,xs:string,the max length is 256--></Password>
    <Protocol><!--req,xs:string,http or ftp--></Protocol>
    <Path><!--req,xs:string,the max length is 128--></Path>
</CloundUpgradeServer>
```

9.2.34 /System/CloudUpgradeFirmwareList

/System/CloudUpgradeFirmwareList	General Resource v1.0
GET	Viewer
Description	It is used to get cloud upgrade firmware list.
Query	None
Inbound Data	**FirmwareList**
Success Return	**ResponseStatus**
Notes: Example: http://192.168.1.11/System/CloudUpgradeFirmwareList?VersionType=all | default (all, It is get all firmware version list, default,It is get a the newest firmware version)	

FirmwareList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<FirmwareList><!--req-->
```

```
    <Firmware>
      <Name><!--req,xs:string--></Name>
      <Desc><!--req,xs:string--></Desc>
      <Version><!—req,xs:string--></Version>
      <TotalSize><!--req,xs:interger--></TotalSize>
  </Firmware>
</FirmwareList>
```

9.2.35 /System/CloudUpgradeDownloadStart

/System/CloudUpgradeDownloadStart	General Resource v1.0
PUT	Viewer
Description	It is used to Start the Download.
Query	None
Inbound Data	**Action**
Success Return	**ResponseStatus**
Notes: <Passport> It must be a unique string and generated by the client, and all subsequent operations depend on this string. <FirmwareName> It is a string getting by the url "/System/CloudUpgradeFirmwareList"	

Action XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Action><!--req-->
    <Passport><!--req,xs:string--></Passport>
    <FirmwareName><!--req,xs:string,--></FirmwareName>
</Action>
```

9.2.36 /System/CloudUpgradeDownloadStatus

/System/CloudUpgradeDownloadStatus	General Resource v1.0
GET	Viewer
Description	It is used to get the result of the download.
Query	None
Inbound Data	**None**
Success Return	**Status**
Notes: Example: http://192.168.1.11/System/CloudUpgradeDownloadStatus?Passport=abc (the "abc" is a a string generated by the client)	

Status XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Status><!--req-->
        <Status><!--req,xs:string--></Status>
      <TotalSize><!--req,xs:string--></TotalSize>
      <SentSize><!—req,xs:string--></SentSize>
      <Passport><!—req,xs:string--></Passport>
</Status>
```

9.2.37 /System/CloudUpgradeDownloadTerminate

/System/CloudUpgradeDownloadTerminate	General Resource v1.0
PUT	Viewer
Description	It is used to terminate the Download.
Query	None
Inbound Data	**Action**
Success Return	**ResponseStatus**
Notes:	

Action XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Action><!--req-->
     <Passport><!--req,xs:string--></Passport>
</Action>
```

9.2.38 /System/CloudUpgradeStartUpgrade

/System/CloudUpgradeStartUpgrade	General Resource v1.0
PUT	Viewer
Description	It is used to start to upgrade.
Query	None
Inbound Data	**Action**
Success Return	**ResponseStatus**
Notes:	

Action XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Action><!--req-->
      <Passport><!--req,xs:string--></Passport>
```

```
</Action>
```

9.2.39 /System/CloudUpgradeResult

/System/CloudUpgradeResult	General Resource v1.0
GET	Viewer
Description	It is used to get the result of the cloud upgrade
Query	None
Inbound Data	**None**
Success Return	**UpgradeStatus**
Notes: Example: http://192.168.1.11/System/CloudUpgradeResult?Passport=abc(the "abc" is a a string generated by the client)	

UpgradeStatus XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<UpgradeStatus><!--req-->
        <Status><!--req,xs:string--></Status>
        <Passport><!--req,xs:string--></Passport>
        <FirmwareName><!—req,xs:string--></FirmwareName>
</UpgradeStatus>
```

9.2.40 /System/ErrorInfoMap

/System/ErrorInfoMap	General Resource v1.0
GET	Viewer
Description	It is used to get the special error description.
Query	None
Inbound Data	**None**
Success Return	**ErrorInfo**
Notes: <ErrorCode> it is a returnd error code by the http server. <ErrorDescription>it is a returnd error description by the http server. Example: http://192.168.1.11/System/ErrorInfoMap?ErrorCode=(0|1|2|....)	

ErrorInfo XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<ErrorInfo><!--req-->
```

```
        <ErrorCode><!--req,xs:interger--></ErrorCode>
        <ErrorDescription><!--req,xs:string--></ErrorDescription>
</ErrorInfo>
```

9.2.41 /System/DeviceCap

/System/DeviceCap	General Resource v1.0
GET	Viewer
Description	It is used to get the device capability.
Query	None
Inbound Data	**None**
Success Return	**Device**
Notes: With the upgrading of devices and the increase of functions, the following XML is constantly updated.	

Device XML Block

```
<?xml version="1.0" encoding="UTF-8" ?>
<Device>
    <AlarmOutCount>0</AlarmOutCount>
    <AlarmInCount>0</AlarmInCount>
    <FunctionList>
        <SD><!--xs:boolean--></SD>
        <WIFI><!--xs:boolean--></WIFI>
        <TwoWayAudio><!--xs:boolean--></ TwoWayAudio >
        <AP><!--xs:boolean--></AP>
        <Wavelink><!--xs:boolean--></Wavelink>
        <Smartlink><!--xs:boolean--></Smartlink>
        <Email><!--xs:boolean--></Email>
        <FTP><!--xs:boolean--></FTP>
        <RemoteRebooe><!--xs:boolean--></RemoteReboot>
        <Restore><!--xs:boolean--></Restore>
        <PasswordModify><!--xs:boolean--></PasswordModify>
    </FunctionList>
    <FunctionListAboutChannel>
        <ChannelList>
            <Channel ChannelID="0">
                <Audio><!--xs:boolean--></Audio>
                <Motion><!--xs:boolean--></Motion>
                <ColorNight><!--xs:boolean--></ColorNight>
                <IntelligentNight><!--xs:boolean--></IntelligentNight>
                <MotionSenstive>
```

```
                    <Highest><!--xs:boolean--></Highest>
                    <Higher><!--xs:boolean--></Higher>
                    <High><!--xs:boolean--></High>
                    <Middle><!--xs:boolean--></Middle>
                    <Low><!--xs:boolean--></Low>
                    <Lower><!--xs:boolean--></Lower>
                    <Lowest><!--xs:boolean--></Lowest>
                </MotionSenstive>
                <multi_osd>true</multi_osd>
                <mosd_cnt>16</mosd_cnt>
                <mosd_cord>16</mosd_cord>
            </Channel>
        </ChannelList>
    </FunctionListAboutChannel>
</Device>
```

9.2.42 /System/AudioAlarmConfig

/System/AudioAlarmConfig	General Resource v1.0
GET	Viewer
Description	Get the sound alarm configuration
Query	None
Inbound Data	**None**
Success Return	**AudioAlarm**
PUT	Viewer
Description	Set the sound alarm configuration
Query	None
Inbound Data	**AudioAlarm**
Success Return	**ResponseStatus**
Notes:	

AudioAlarm XML

```
<?xml version="1.0" encoding="utf-8"?>
<AudioAlarm><!--req-->
    <Enable><!--req,xs:boolean--></Enable>
    <Type><!--req,xs:string,“custom,alarm,dog,warning area,monitoring area,welcome,Deep Water Warning,Private territory,danger zone,Valuable objects,High Warning,Private Parking,Ding Dong,HasMask,NoMask”--></Type>
    <Delay><!--req,xs:interger,the unit is second--></Delay>
    <Volume><!--req,xs:interger,0--100--></Volume>
</AudioAlarm>
```

9.2.43 /System/TryListenSelectedAlarmAudio

/System/TryListenSelectedAlarmAudio	General Resource v1.0
PUT	Viewer
Description	Listen to the currently selected alarm audio
Query	None
Inbound Data	**AudioAlarm**
Success Return	**ResponseStatus**
Notes:	

AudioAlarm XML

```
<?xml version="1.0" encoding="utf-8"?>
<AudioAlarm><!--req-->
        <Enable><!--req,xs:boolean--></Enable>
        //custom, alarm, dog, warning area, monitoring area, welcome, Deep Water Warning, Private territory, danger zone, Valuable objects, High Warning,Private Parking
       <Type><!--req,xs:string,“alarm,dog,custom”--></Type>
        <Delay><!--req,xs:interger,the unit is second--></Delay>
        <Volume><!--req,xs:interger,0--100--></Volume>
</AudioAlarm>
```

9.2.44 /System/SaveHardwareConfig

/System/SaveHardwareConfig	General Resource v1.0
PUT	Viewer
Description	Save hardware configuration
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.2.45 /System/CancelHardwareConfig

/System/CancelHardwareConfig	General Resource v1.0
PUT	Viewer
Description	Unconfigure hardware
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

9.2.45 /System/AuthInfo

/System/AuthInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the activation info
Query	None
Inbound Data	**None**
Success Return	**AuthInfo**
PUT	Viewer
Description	It is used to activate or unactivate the device
Query	None
Inbound Data	**AuthInfo**
Success Return	**ResponseStatus**
Notes: **If you want to unactivate the device, set the "AuthInfo" as empty**	

AuthInfo XML

```
<Auth>
    <AuthInfo><!--req,xs:string--></AuthInfo >
</Auth
```

9.2.46 /System/DeviceCMS

/System/DeviceCMS	General Resource v1.0
GET	Viewer
Description	It is used to get the config of center server
Query	None
Inbound Data	**None**
Success Return	cms
PUT	Viewer
Description	It is used to set the config of center server
Query	None
Inbound Data	**cms**
Success Return	**ResponseStatus**
Notes:	

cms XML

```
<cms>
    <Enable><!--req,xs:boolean--></Enable >
    <ServerAddress><!--req,xs:string--></ServerAddress>
    <Port><!--req,xs:interger--></Port>
    <RegisterInterval><!--req,xs:interger--></ RegisterInterval>
</cms
```

9.2.47 /System/DeviceInfoV2

/System/DeviceInfoV2	General Resource v1.0
GET	Viewer
Description	It is used to get the device info v2
Query	None
Inbound Data	**None**
Success Return	ResponseStatus or DeviceInfoV2
PUT	Operator
Description	It is used to set the device info v2
Query	None
Inbound Data	**DeviceInfoV2**
Success Return	**ResponseStatus**
Notes:	

DeviceInfoV2 XML

```
<DeviceInfoV2>
    <SerialCustom><!--opt,xs:boolean--></SerialCustom >
    <SerialNumber><!--req,xs:string--></SerialNumber>
</DeviceInfoV2>
```

9.2.48 /System/DeviceReversePacket

PUT	Operator
Description	It is used to control device to make update packet and save to storage.
Query	None
Inbound Data	**DeviceInfoV2**
Success Return	**ResponseStatus**
Notes:	

9.2.49 /System/ChannelID/AssistantInterfaceControl

/System/ChannelID/AssistantInterfaceControl	General Resource v1.0
GET	Viewer
Description	It is used to get the status of assistant interface
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AssisstantInterfaceList**
PUT	**Operator**
Description	It is used to set the status of assistant interface
Query	None
Inbound Data	**AssisstantInterfaceList**
Success Return	**ResponseStatus**
Notes: **The status will recovery after restart.**	

AssisstantInterfaceList XML

```
<AssisstantInterfaceList>
    <AssisstantInterface>
        <ChannelID><!--req,xs:Interger--></ChannelID>      //no need when put
        <AssisstantID><!--req,xs:Interger--></AssisstantID>
        <Status><!--req,xs:string"Disable","Enable"--></Status>
    </AssisstantInterface>
    …
</AssisstantInterfaceList>
```

9.2.50 /System/WorkModeControl

/System/WorkModeControl	General Resource v1.0
GET	Viewer
Description	It is used to get the work mode of system
Query	None
Inbound Data	None
Success Return	**ResponseStatus or SystemWorkMode**
PUT	**Operator**
Description	It is used to set the work mode of system
Query	None
Inbound Data	**SystemWorkMode**
Success Return	**ResponseStatus**
Notes: **The work mode will recovery after restart**	

SystemWorkMode XML

```
<SystemWorkMode>
    <WorkMode><!--req,xs:string"TestMode","NormalMode"--></WorkMode>
</SystemWorkMode>
```

9.2.51 /System/DeviceBindConfig

/System/DeviceBindConfig	General Resource v1.0
GET	Viewer
Description	It is used to get the config of device bind
Query	None
Inbound Data	None
Success Return	**ResponseStatus or DeviceBindConfig**
PUT	**Operator**
Description	It is used to set the config of device bind
Query	None
Inbound Data	**DeviceBindConfig**
Success Return	**ResponseStatus**
Notes: **SCodeValid is not required in "HTTP Put"**	

DeviceBindConfig XML

```
<DeviceBindConfig>
    <SCodeValid><!--ro,opt,xs:boolean"true","false"--></SCodeValid>
    <SCode><!--rw,req,xs:string--></SCode>
</DeviceBindConfig>
```

9.2.52 /System/LongSeeDeviceConfig

/System/LongSeeDeviceConfig	General Resource v1.0
GET	Viewer
Description	It is used to get the config of longsee device
Query	None
Inbound Data	None
Success Return	**ResponseStatus or LongSeeDeviceInfo**
PUT	**Operator**
Description	It is used to set the config of longsee device
Query	None
Inbound Data	**LongSeeDeviceInfo**
Success Return	**ResponseStatus**
Notes:	

LongSeeDeviceInfo XML

```
<LongSeeDeviceInfo>
    <MacAddress><!--rw,req,xs:string--></MacAddress>
    <SerialNum><!--rw,req,xs:string--></SerialNum>
</LongSeeDeviceInfo>
```

9.2.53 /System/DeleteVoiceCustom

/System/DeleteVoiceCustom	General Resource v1.0
PUT	**Operator**
Description	It is used to delete voice custom
Query	None
Inbound Data	**LongSeeDeviceInfo**
Success Return	**ResponseStatus**
Notes:	

9.2.54 /System/Language

/System/Language	General Resource v1.0
GET	Viewer
Description	It is used to get the system language
Query	None
Inbound Data	None
Success Return	**ResponseStatus or SystemLanguageConfig**
PUT	**Operator**
Description	It is used to set the system language
Query	None
Inbound Data	**SystemLanguageConfig**
Success Return	**ResponseStatus**
Notes:	

SystemLanguageConfig XML

```
<SystemLanguageConfig>
    <Language><!--rw,req,xs:string--></Language>
</SystemLanguageConfig>
```

9.2.55 /System/PerformanceStatus

/System/PerformanceStatus	General Resource v1.0
GET	Viewer

Description	It is used to get the system performance status
Query	None
Inbound Data	None
Success Return	**ResponseStatus or SystemPerformanceStatus**
Notes:	

SystemPerformanceStatus XML

```
<SystemPerformanceStatus>
    <CPUPercent><!--req,xs:string”80%”--></CPUPercent>
    <MemoryPercent><!--req,xs:string”80%”--></MemoryPercent>
    <StroagePercent><!--req,xs:string”80%”--></StroagePercent>
    <UpTime><!--req,xs:Interger--></UpTime>
    <OpenDoorCount><!--req,xs:Interger--></OpenDoorCount>
    <BootCount><!--req,xs:Interger--></BootCount>
    <MemorySize><!--req,xs:Interger--></MemorySize>
</SystemPerformanceStatus>
```

9.2.56 /System/FiberHomeBaseInfo

/System/FiberHomeBaseInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the base info of camera
Query	None
Inbound Data	None
Success Return	**ResponseStatus or CameraBaseInfo**
Notes:	

CameraBaseInfo XML

```
<CameraBaseInfo>
    <ManufactureID><!--req,xs:string--></ManufactureID>
    <DevMod><!--req,xs:string--></DevMod>
    <DetailID><!--req,xs:string eg: 13001SN-01--></DetailID>
</CameraBaseInfo>
```

9.2.57 /System/FiberHomeLocationInfo

/System/FiberHomeLocationInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the location info
Query	None
Inbound Data	None
Success Return	**ResponseStatus or CameraLocationInfo**

PUT	Operator
Description	It is used to set the location info
Query	None
Inbound Data	**CameraLocationInfo**
Success Return	**ResponseStatus**
Notes:	

CameraLocationInfo XML

```
<CameraLocationInfo>
    <SiteName><!--rw,req,xs:string--></SiteName>
    <SiteAddr><!--rw,req,xs:string--></SiteAddr>
    <DevName><!--rw,req,xs:string--></DevName>
    <DevAddr><!--rw,req,xs:string--></DevAddr>
</CameraLocationInfo>
```

9.2.58 /System/FiberHomeCtrlChnlServer

/System/FiberHomeCtrlChnlServer	General Resource v1.0
GET	Viewer
Description	It is used to get ctrl chnl server
Query	None
Inbound Data	None
Success Return	**ResponseStatus or CtrlChnlServerInfo**
PUT	**Operator**
Description	It is used to set ctrl chnl server
Query	None
Inbound Data	**CtrlChnlServerInfo**
Success Return	**ResponseStatus**
Notes:	

CtrlChnlServerInfo XML

```
<CtrlChnlServerInfo>
    <CtrlChnlServer><!--rw,req,xs:string--></CtrlChnlServer>
</CtrlChnlServerInfo>
```

9.2.59 /System/FiberHomeLocationID

/System/FiberHomeLocationID	General Resource v1.0
GET	Viewer
Description	It is used to get location id
Query	None
Inbound Data	None

Success Return	**ResponseStatus or LocationIDInfo**
PUT	**Operator**
Description	It is used to set the location id
Query	None
Inbound Data	**LocationIDInfo**
Success Return	**ResponseStatus**
Notes:	

LocationIDInfo XML

```
<LocationIDInfo>
    <LocationID><!--rw,req,xs:string--></LocationID>
</LocationIDInfo>
```

9.2.60 /System/FiberHomeDataCenterServer

/System/FiberHomeDataCenterServer	General Resource v1.0
GET	Viewer
Description	It is used to get the date center server
Query	None
Inbound Data	None
Success Return	**ResponseStatus or DataCenterServerInfo**
PUT	**Operator**
Description	It is used to set the date center server
Query	None
Inbound Data	**DataCenterServerInfo**
Success Return	**ResponseStatus**
Notes:	

DataCenterServerInfo XML

```
<DataCenterServerInfo>
    <DataCenterServer><!--rw,req,xs:string--></DataCenterServer>
</DataCenterServerInfo>
```

9.2.61 /System/FiberHomeAreaCodeInfo

/System/FiberHomeAreaCodeInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the area code
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AreaCodeInfo**

PUT	Operator
Description	It is used to set the area code
Query	None
Inbound Data	**AreaCodeInfo**
Success Return	**ResponseStatus**
Notes:	

AreaCodeInfo XML

```
<AreaCodeInfo>
    <AreaCode><!--rw,req,xs:string--></AreaCode>
</AreaCodeInfo>
```

9.2.62 /System/MQTTServerConfig

/System/MQTTServerConfig	General Resource v1.0
GET	Viewer
Description	It is used to get the MQTT server config
Query	None
Inbound Data	None
Success Return	**ResponseStatus or MQTTServerConfig**
PUT	**Operator**
Description	It is used to set the MQTT server config
Query	None
Inbound Data	**MQTTServerConfig**
Success Return	**ResponseStatus**
Notes:	

MQTTServerConfig XML

```
<MQTTServerConfig>
    <Enable><!--rw,xs:boolean--></Enable>
    <DeviceID><!--rw,xs:string,the max length is 64--></DeviceID>
<ServerAddress><!--rw,xs:string,domain or ip,the max length is 128--> </ServerAddress>
<ServerPort><!--rw,xs:interger--></ServerPort>
<Direction><!--rw,xs:string,"none,enter,exit"--></Direction>
<ServerUserName><!--rw,xs:string,the max length is 64--></ServerUserName>
<ServerPassword><!--rw,xs:string,the max length is 64--></ServerPassword>
<ConnectStatus><!-ro,xs:string,”init,success,erro_userorpassword,error_id_conflict,error_networ
k_except ion,unknown"--></ConnectStatus>
    <MainTopic><!--rw,xs:string,the max length is 64--></MainTopic>
    <OperateDeviceTopic><!--ro,xs:string,the max length is 128--> </OperateDeviceTopic>
    <PublicEventTopic><!--ro,xs:string,the max length is 128--> </PublicEventTopic>
</MQTTServerConfig>
```

9.2.63 /System/AIWorkMode

/System/AIWorkMode	General Resource v1.0
GET	Viewer
Description	It is used to get the AI algorithm work mode
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AIWorkMode**
PUT	**Operator**
Description	It is used to set the AI algorithm work mode
Query	None
Inbound Data	**AIWorkMode**
Success Return	**ResponseStatus**
Notes:	

AIWorkMode XML

```
<AIWorkMode>
    <WorkMode><!--req,xs:string “PeopleMode”,”FaceMode”--></WorkMode>
</AIWorkMode>
```

9.3 /Security

9.3.1 /Security/UserAuth

/Security/UserAuth	General Resource v1.0
PUT	Viewer
Description	It is used to verify user information
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

You not need to set the http User authentication information.

Example:

URL: http://192.168.1.46/Security/UserAuth

Data format:

```
PUT /Security/UserAuth HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Username=admin&Password=admin
```

9.3.2 /Security/ModifyPasswordByTempSecretKey

/Security/ModifyPasswordByTempSecretKey	General Resource v1.0
PUT	Operator
Description	It is used to modify password by temp secret key.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

9.3.3 /Security/BaseConfig

/Security/BaseConfig	General Resource v1.0
GET	Viewer
Description	It is used to get the security config
Query	None
Inbound Data	None
Success Return	**ResponseStatus or BaseConfig**
PUT	**Operator**
Description	It is used to set the security config
Query	None
Inbound Data	**BaseConfig**
Success Return	**ResponseStatus**
Notes:	

BaseConfig XML

```
<BaseConfig>
    <WhiteList><--xs:boolean--></WhiteList>
</BaseConfig>
```

9.3.4 /Security/WhiteLists

/Security/WhiteLists	General Resource v1.0
GET	Viewer
Description	It is used to get white list
Query	None
Inbound Data	None
Success Return	**ResponseStatus or WhiteLists**
Notes:	

WhiteLists XML

```
<WhiteLists>
```

```
    <WhiteList>
        <ID><!--xs:integer,the value is unique--></ID>
        <Mode><!--req,xs:string,"IP","IPRange" --></Mode>
        //Mode:IP
        <IP><!--xs:string, --></IP>
        //Mode"IPRange
        <IPBegin><!--xs:string, --></IPBegin>
        <IPEnd><!--xs:string, --></IPEnd>
    </WhiteList>
</WhiteLists>
```

eg:

```
<WhiteLists>
    <WhiteList>
        <ID>1</ID>
        <Mode>IP</Mode>
        <IP>192.168.0.10</IP>
    </WhiteList>
    <WhiteList>
        <ID>2</ID>
        <Mode>IPRange</Mode>
        <IPBegin>192.168.0.10</IPBegin>
        <IPEnd>192.168.0.20</IPEnd>
    </WhiteList>
</WhiteLists>
```

9.3.5 /Security/WhiteList

/System/WhiteList	General Resource v1.0
GET	Viewer
Description	It is used to get the white list
Query	None
Inbound Data	None
Success Return	**WhiteList**
PUT	**Operator**
Description	It is used to update the white list
Query	None
Inbound Data	**WhiteList**
Success Return	**ResponseStatus**
POST	**Operator**
Description	It is used to add a item of white list
Query	None
Inbound Data	**WhiteList**
Success Return	**ResponseStatus**

Delete	Operator
Description	It is used to delete the whilte list
Query	None
Inbound Data	**WhiteList**
Success Return	**ResponseStatus**
Notes：	

WhiteList XML

```
<WhiteList>
    <ID><!--xs:integer,the value is unique--></ID>
    <Mode><!--req,xs:string,"IP","IPRange" --></Mode>
    //Mode：IP
    <IP><!--xs:string, --></IP>
    //Mode：IPRange
    <IPBegin><!--xs:string, --></IPBegin>
    <IPEnd><!--xs:string, --></IPEnd>
</WhiteList>
```

eg input：

```
<WhiteList>
    <ID>1</ID>
</WhiteList>
```

Eg 1：

```
<WhiteList>
    <ID>1</ID>
    <Mode>IP</Mode>
    <IP>192.168.0.10</IP>
</WhiteList>
```

Eg 2：

```
<WhiteList>
    <ID>1</ID>
    <Mode>IPRange</Mode>
    <IPBegin>192.168.0.10</IPBegin>
    <IPEnd>192.168.0.20</IPEnd>
</WhiteList>
```

Delete eg，input：

```
<WhiteList>
    //if has id, then delete the white item by the id specified, otherwise delete all
    <ID>1</ID>
</WhiteList>
```

Add eg，input：

```
<WhiteList>
    //no need
    <ID>2</ID>
    <Mode>IP</Mode>
    <IP>192.168.0.30</IP>
```

```
</WhiteList>
```

Update eg，input：

```
<WhiteList>
    <ID>2</ID>
    <Mode>IP</Mode>
    <IP>192.168.0.60</IP>
</WhiteList>
```

Update eg，output：

```
<WhiteList>
    <ID>2</ID>
    <Mode>IPRange</Mode>
    <IPBegin>192.168.0.30</IPBegin>
    <IPEnd>192.168.0.40</IPEnd>
</WhiteList>
```

9.3.6 /Security/UserPermissionConfigs

/Security/UserPermissionConfigs	General Resource v1.0
GET	Viewer
Description	It is used to get the user permission
Query	None
Inbound Data	None
Success Return	**ResponseStatus or UserBasicConfigList**
PUT	**Operator**
Description	设置安全配置
Query	None
Inbound Data	**UserBasicConfigList**
Success Return	**ResponseStatus**
Notes：	

UserBasicConfigList XML

```
<UserBasicConfigList>
    <UserBasicConfig>
        <UserID><!--rw,req,xs:Interger --></UserID>
        <Username><!--ro,req,xs:string--></Username>        //No need on put
    <UserType><!--rw,req,xs:string”Administrator”,”DefaultUser”,”Normal”,”OnvifUser”,”OnvifOperator”--></UserType>
    <RemoteConfigPermission><!--rw,req,xs:boolean --></RemoteConfigPermission>
    <RealtimeStreamPermission><!--rw,req,xs:string”1”,”0”--></RealtimeStreamPermission>
    <VoiceTalkPermission><!--rw,req,xs:boolean--></VoiceTalkPermission>
    <PTZControlPermission><!--rw,req,xs:string”1”,”0”--></PTZControlPermission>
    <RemotePlaybackPermission><!--rw,req,xs:string”1”,”0”--></RemotePlaybackPermission>
    </UserBasicConfig>
```

```
    …
</UserConfigList>
```

9.3.7 /Security/UserPermissionConfig/UserID

/Security/UserPermissionConfig/UserID	General Resource v1.0
GET	Viewer
Description	It is used to get the user permission
Query	None
Inbound Data	None
Success Return	**ResponseStatus or UserBasicConfig**
PUT	**Operator**
Description	It is used to set the user permission
Query	None
Inbound Data	**UserBasicConfig**
Success Return	**ResponseStatus**
Notes: **UserID start from 1**	

UserBasicConfig XML

```
<UserBasicConfig>
    <UserID><!--rw,req,xs:Interger --></UserID>        //No need on put
    <Username><!--ro,req,xs:string--></Username>          //No need on put
    <UserType><!--rw,req,xs:string”Administrator”,”DefaultUser”,”Normal”,”OnvifUser”,”OnvifOp
erator”--></UserType>
    <RemoteConfigPermission><!--rw,req,xs:boolean --></RemoteConfigPermission>
    <RealtimeStreamPermission><!--rw,req,xs:string”1”,”0”--></RealtimeStreamPermission>
    <VoiceTalkPermission><!--rw,req,xs:boolean--></VoiceTalkPermission>
    <PTZControlPermission><!--rw,req,xs:string”1”,”0”--></PTZControlPermission>
    <RemotePlaybackPermission><!--rw,req,xs:string”1”,”0”--></RemotePlaybackPermission>
</UserBasicConfig>
```

9.3.8 /Security/UserPermissionConfigByUsername

/Security/UserPermissionConfigByUsername	General Resource v1.0
GET	Viewer
Description	It is used to get the user permission
Query	None
Inbound Data	Username=xxx
Success Return	**ResponseStatus or UserBasicConfig**
PUT	**Operator**

Description	It is used to set the user permission
Query	None
Inbound Data	**UserBasicConfig**
Success Return	**ResponseStatus**
Notes: **URLeg:** **http://xxx.xxx.xxx.xxx/Security/UserPermissionConfigByUsername?Username=xxx**	

UserBasicConfig XML

```
<UserBasicConfig>
    <UserID><!--rw,req,xs:Interger --></UserID>        //No need on put
    <Username><!--ro,req,xs:string--></Username>          // No need on put
    <UserType><!--rw,req,xs:string”Administrator”,”DefaultUser”,”Normal”,”OnvifUser”,”OnvifOp
erator”--></UserType>
    <RemoteConfigPermission><!--rw,req,xs:boolean --></RemoteConfigPermission>
    <RealtimeStreamPermission><!--rw,req,xs:string”1”,”0”--></RealtimeStreamPermission>
    <VoiceTalkPermission><!--rw,req,xs:boolean--></VoiceTalkPermission>
    <PTZControlPermission><!--rw,req,xs:string”1”,”0”--></PTZControlPermission>
    <RemotePlaybackPermission><!--rw,req,xs:string”1”,”0”--></RemotePlaybackPermission>
</UserBasicConfig>
```

9.4 /Streams

/Streams	General Resource v1.0
GET	Viewer
Description	It is used to get the properties of streaming channels for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or StreamList**
PUT	Operator
Description	It is used to update the properties of streaming channels for the device.
Query	None
Inbound Data	**StreamList**
Success Return	**ResponseStatus**
Notes:	

Streams XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<StreamList>
    <Stream><!-- opt -->
```

```
</StreamList>
```

9.4.1 /Streams/ChannelID

/Streams/ChannelID	General Resource v1.0
GET	Viewer
Description	It is used to get the properties of a particular streaming channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Stream**
PUT	Viewer
Description	It is used to set the properties of a particular streaming channel for the device.
Query	None
Inbound Data	**Stream**
Success Return	**ResponseStatus**
Notes: The ChannelID in URI shows the channel number. The <ChannelID> is start at 1.	

Stream XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Stream>
    <ID><!--opt,xs:integer--></ID>
    <StreamCapability><!--opt-->
    <StreamConfigList>
        <StreamConfig><!--opt-->
    </StreamConfigList>
</Stream>
```

9.4.1.1 /Streams/ChannelID/ReceiveData

/Streams/ChannelID/StreamID/ReceiveData	General Resource v1.0
GET	Viewer
Description	It is used to get the MJPEG stream for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Stream data**
Notes: The channelID in the URI shows the channel number, The <ChannelID> is start at 1.defaultly	

the first substream is opened.

9.4.1.2 /Streams/ChannelID/CapabilityV2

/Streams/ChannelID/CapabilityV2	General Resource v1.0
GET	Viewer
Description	It is used to get the Stream Capability of a particular streaming channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or StreamCapability**

Notes:

The <ChannelID> in the URI shows the channel number, The <ChannelID> is start at 1.

The <ID> under the <StreamCapability> represents the channel ID.

The <ID> under the <MainStreamCapability> indicates the main stream of ID the device supported

The <ID> under the <MainStreamCapability>,<SecondStreamCapabilityList>,

indicates the sub stream of ID the device supported Resolution and maximum frame rate supported by the main stream is not the same, but different main stream supports different sub-stream: Understanding the structure of the flow capacity,It is not the same, that is to say the resolution and maximum frame rate of the sub-stream by the resolution of the main stream to the decision. Therefore, the design follows the structure to represent the encoding capabilities of the device. For example, a device supports main stream 720P / 25fps and 1080P / 25fps, while the 720P erupted stream supported resolution and maximum frame rate is 360p / 20fps, 1080p stream erupted in support of the resolution and maximum frame rate is D1 / 25fps.

<CodecCapabilityList> describe which codec type support.such as H.264,H265,MJPEG,H.265+,H264+.

StreamCapability XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<StreamCapability>
        <ID><!--req,xs:integer, --></ID>
      <MainStreamCapabilityList><!--ro,req-->
            <MainStreamCapability><!---ro,opt->
                  <ID><!--req,xs:integer, --></ID>
                  <ResolutionWidth><!--ro,opt,xs:integer--></ResolutionWidth>
                  <ResolutionHeigth><!--ro,opt,xs:integer--></ResolutionHeigth>
                  <MaxFrameRate><!--ro,req,xs:integer,--></MaxFrameRate>
                  <SubStreamCapabilityList><!--ro,req-->
                        <StreamCapability><!--ro,opt-->
                              <ID><!--ro,req,xs:integer--></ID>
                              <ResolutionWidth><!--ro,opt,xs:integer--></ResolutionWidth>
```

```
                <ResolutionHeigth><!--ro,opt,xs:integer--></ResolutionHeigth>
                <MaxFrameRate><!--ro,req,xs:integer,--></MaxFrameRate>
            </StreamCapability>
        </SubStreamCapabilityList>
        <SecondStreamCapabilityList><!--ro,req-->
            <StreamCapability><!--ro,opt-->
                <ID><!--ro,req,xs:integer--></ID>
                <ResolutionWidth><!--ro,opt,xs:integer--></ResolutionWidth>
                <ResolutionHeigth><!--ro,opt,xs:integer--></ResolutionHeigth>
                <MaxFrameRate><!--ro,req,xs:integer,--></MaxFrameRate>
            </StreamCapability>
        </SecondStreamCapabilityList>
    </MainStreamCapability>
    <CodecCapabilityList Version="1.0">
        <streamCodecCapability Version="1.0">
            <ID><!--ro,req,xs:integer--></ID>
            <Codec><!--req,xs:string,"H.264,H.265,MJPEG,H.265+,H.264+"--></Codec>
        </streamCodecCapability>
    </CodecCapabilityList>
    </MainStreamCapabilityList>
</StreamCapability>
```

9.4.1.3 /Streams/ChannelID/StreamID

/Streams/ChannelID/StreamID	General Resource v1.0
GET	Viewer
Description	It is used to get the StreamID Streaming properties of a particular streaming channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or StreamConfig**
PUT	Viewer
Description	It is used to set the StreamID Streaming properties of a particular streaming channel for the device.
Query	None
Inbound Data	**StreamConfig**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in the URI shows the channel number, The <ChannelID> is start at 1. The <StreamID> represents stream type, generally "1" represents the main stream, "2" means	

"the first sub-Stream", “3” means “the second sub-stream”, and so on.

The <FrameRate> field indicates the frame rate, if the value is 25500, which indicates that the frame rate is a full frame rate.

The <FixedQuality> field represents the quality, the smaller the value, the higher the quality.

If <StreamID> corresponds to a sub-stream, then <StreamType>, <VideoCodecType>, <FixedQuality> is invalid, <BitrateType> is always a CBR (constant bit rate).

<keyFrameInterval> is not supported.

Resolution and frame rate ranges refer to the <StreamCapability>

If <StreamID> corresponds to a snapshot stream, then <StreamType>, <VideoCodecType>, <FixedQuality> is invalid, <BitrateType> is always a CBR (constant bit rate).

StreamConfig XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<StreamConfig><!--opt-->
    <ID><!--req,xs:integer--><ID>
    <StreamType><!--req,xs:string,”Video,VideoAndAudio”--></StreamType>

<VideoCodecType><!—ro,opt,xs:string,”H.264,H.265,MJPEG,H.265+,H.264+”--></VideoCodecTyp
e>
    <ResolutionWidth><!--req,xs:integer--></ResolutionWidth>
    <ResolutionHeigth><!--req,xs:integer--></ResolutionHeigth>
    <FrameRate><!--req,xs:integer, maximum frame rate x100--></FrameRate>
    <BitrateType><!--req,xs:string,"CBR,VBR,SmartAVC"--></BitrateType>
    <Bitrate><!--opt, xs:integer, in kbps--></Bitrate>
    <FixedQuality><!--req,xs:integer,"0-5"--></FixedQuality>
    <keyFrameInterval> <!-- opt, xs:integer--> </keyFrameInterval>
    <Transport><!--opt-->
        <RTSPURI><!--ro,req,xs:string--></RTSPURI>
    </Transport>
</StreamConfig>
```

9.4.1.3.1 /Streams/ChannelID/StreamID/Transport

/Streams/ChannelID/StreamID/Transport	General Resource v1.0
GET	Viewer
Description	It is used to get the RTSP Address for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Transport**
Notes:	

The <ChannelID> in the URI shows the channel number, The <ChannelID> is start at 1.

The <StreamID> represents stream type, generally "1" represents the main stream, “2” means "the first sub-Stream", and so on.

The <RTSPURI> field indicates the address which is access to the stream by the RTSP Protocol.

Stream Transport XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Transport>
    <RTSPURI><!--ro,req,xs:string--></RTSPURI>
</Transport>
```

9.4.1.3.2 /Streams/ChannelID/StreamID/ReceiveData

/Streams/ChannelID/StreamID/ReceiveData	General Resource v1.0
GET	Viewer
Description	It is used to get the MJPEG stream for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Stream Data**
Notes: The <ChannelID> in the URI shows the channel number, The <ChannelID> is start at 1. The <StreamID> represents stream type, generally "1" represents the main stream, “2” means "the first sub-Stream", and so on.	

9.4.2 /Streams/AudioStream

/Streams/AudioStream	General Resource v1.0
GET	Viewer
Description	It is used to get audio stream configuration.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AudioStream**
PUT	Viewer
Description	It is used to set audio stream configuration.
Query	None
Inbound Data	**AudioStream**
Success Return	**ResponseStatus**
Notes:	

Audio XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AudioStream>
   <AudioInMethod><!—rw,xs:string,”microphone,line-in”--></AudioInMethod>
   <AudioInCodec><!—rw,xs:string,”G.711U”--></AudioInCodec>
   <AudioInVolume><!—rw,xs:integer,”0--100”--></AudioInVolume>
   <AudioOutVolume><!—rw,xs:integer,”0--100”--></AudioOutVolume>
</AudioStream>
```

9.5 /Snapshot

9.5.1 /Snapshot/ChannelID/RemoteImageCapture

/Snapshot/ChannelID/RemoteImageCapture	General Resource v1.0
GET	Viewer
Description	It is used to get a jpeg picture.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or the JPEG picture stream**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

“ImageFormat” is image Type such as,bmp,jpg and so on,if “ImageFormat” is 1,the picture you capture is the format of bmp, if “ImageFormat” is 2, It is the format of jpg, default is the format of jpg, so ,you should set “ImageFormat” to 2, if you set “ImageFormat” to other value,maybe you will get a result you do not want to see.

Example:

http://192.168.1.46/Snapshot/1/RemoteImageCapture?ImageFormat=2

9.5.2 /Snapshot/ChannelID/StreamID/RemoteImageCaptureV2

/Snapshot/ChannelID/StreamID/RemoteImageCaptureV2	General Resource v1.0
GET	Viewer
Description	It is used to get a jpeg picture.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or the JPEG picture stream**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

The <StreamID> represents stream type, generally "1" represents the main stream, “2” means "the first sub-Stream", and so on.

“ImageFormat” is image Type such as,bmp,jpg and so on, if you set “ImageFormat” to other value,maybe you will get a result you do not want to see.

Example:

http://192.168.1.46/Snapshot/1/2/RemoteImageCaptureV2?ImageFormat=bmp|jpg

9.5.3 /Snapshot/Capability

/Snapshot/Capability	General Resource v1.0
GET	Viewer
Description	It is used to get the picture capability.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or SnapshotCapabilityList**
Notes:	

SnapshotConfig XML Block

```
<SnapshotCapabilityList><!--ro,req-->
    <SnapshotCapability><!--ro,opt-->
        <ID><!--ro,req,xs:integer--></ID>
        <ResolutionWidth><!--ro,req,xs:integer--></ResolutionWidth>
        <ResolutionHeight><!--ro,req,xs:integer--></ResolutionHeight>
    </SnapshotCapability>
</SnapshotCapabilityList>
```

9.5.4 /Snapshot/Config

/Snapshot/Config	General Resource v1.0
GET	Viewer
Description	It is used to get a Snapshot Config.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or SnapshotConfig**
PUT	Viewer
Description	It is used to set a Snapshot Config.
Query	None
Inbound Data	**SnapshotConfig**
Success Return	**ResponseStatus**

Notes:

SnapshotConfig XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<SnapshotConfig>
        <ResolutionWidth><!--rw,req,xs:interger--></ResolutionWidth>
        <ResolutionHeight><!--rw,req,xs:interger--></ResolutionHeight>
        <Quality><!—rw,req,xs:string,”high,middle,low” --></Quality>
        <SnapshotCount><!—rw,req,xs:integer,”1,2,3” --></SnapshotCount>
        <SnapshotIntervalTime><!—rw,req,xs:interger,in sec,”1,2,3,4,5” -->
        </ SnapshotIntervalTime>
        <StreamType><!—rw,req,xs:string,”main,sub,hdsub”--></StreamType>
</SnapshotConfig>
```

9.6 /Images

/Images	General Resource v1.0
GET	Viewer
Description	It is used to get All Image Configuration for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or ImageList**
PUT	Viewer
Description	It is used to set All Image Configuration for the device.
Query	None
Inbound Data	**ImageList**
Success Return	**ResponseStatus**
Notes:	

ImageList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<ImageList><!--req-->
      <Image><!--opt-->
</ImageList>
```

9.6.1 /Images/ChannelID

/Images/ ChannelID	General Resource v1.0
GET	Viewer
Description	It is used to get All Image Configuration of a particular channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Image**
PUT	Viewer
Description	It is used to set All Image Configuration of a particular channel for the device.
Query	None
Inbound Data	**Image**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Image XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Image>
    <ID><!--req,xs:integer--></ID>
    <Scene><!--opt-->
    <Basic/><!--opt-->
    <Color/><!--opt-->
    <Exposure/><!--opt-->
    <WhiteBlance/><!--opt-->
    <IrCutFillter><!--opt-->
        //LDC Advance, range: 0-100
        <LDCAdvance><!--req,xs:integer--></LDCAdvance>
    </IrCutFillter>
</Image>
```

9.6.1.1 /Images/ChannelID/Scene

/Images/ChannelID/Scene	General Resource v1.0
GET	Viewer
Description	It is used to get the scene configuration of a particular channel for the device.
Query	None
Inbound Data	None

Success Return	**ResponseStatus or Scene**
PUT	Viewer
Description	It is used to set the scene configuration of a particular channel for the device.
Query	None
Inbound Data	**Scene**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Scene XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Scene>
    <Support><!—ro,req,xs:boolean></Support>
    <Mode><!—req,xs:string,”standard,soft,vivid,custom”></Mode>
    <LDC><!—req,xs:boolean ></LDC>
    <Fog><!—req,xs:boolean ></Fog>
    <FogValue><!--rw,req,xs:interger,0-100--></FogValue>
    <Corridor><!—req,xs:boolean ></Corridor>
    <Gamma><!--rw,req,xs:interger,0-100--></Gamma>
</Scene>
```

9.6.1.2 /Images/ChannelID/Basic

/Images/ChannelID/Basic	General Resource v1.0
GET	Viewer
Description	It is used to get the basic configuration of a particular channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Basic**
PUT	Viewer
Description	It is used to set the basic configuration of a particular channel for the device.
Query	None
Inbound Data	**Basic**
Success Return	**ResponseStatus**
Notes:	

Basic XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Basic>
    <powerLineFrequencyMode><!--req,xs:string,50HZ,60HZ --></ powerLineFrequencyMode>
    <NoiseReduce><!--req,xs:string,"close,weak,normal,strong"--></NoiseReduce>
    <BLC><!--req,xs:string,"close,open,notsupport"--></BLC>
    <WDR><!--req,xs:string, "close,weak,normal,strong"--></WDR>
    <FlipMode>
        <!--req,xs:string,"close,leftright,updown,center,notsupport"-->
    </FlipMode>
</Basic>
```

9.6.1.3 /Images/ChannelID/Color

/Images/ChannelID/Color	General Resource v1.0
GET	Viewer
Description	It is used to get the color configuration of a particular channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Color**
PUT	Viewer
Description	It is used to set the color configuration of a particular channel for the device.
Query	None
Inbound Data	**Color**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

<Mode> shows automatically adjust the brightness.When the mode is "auto", setting the value of the <Brightness> field are unsuccessed, when the mode is "manual", setting the value of the <Brightness> field is valid.For unsupported devices, setting the value of the <Brightness> field is always valid.

Color XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Color>
    <Mode><!--req,xs:string,"auto,manual,notsupport"--></Mode>
    <Brightness><!--req,xs:integer,0-100--></Bright>
    <Contrast><!--req,xs:integer,0-100--></Contrast>
    <Saturation><!--req,xs:integer,0-100--></Saturation>
    <Hue><!--req,xs:integer,0-100--></Hue>
```

```
    <Sharpness><!--req,xs:integer,0-100--></Sharpness>
</Color>
```

9.6.1.4 /Images/ChannelID/Exposure

/Images/ChannelID/Exposure	General Resource v1.0
GET	Viewer
Description	It is used to get the exposure configuration of a particular channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Exposure**
PUT	Viewer
Description	It is used to set the exposure configuration of a particular channel for the device.
Query	None
Inbound Data	**Exposure**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Exposure XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Exposure>
    <IrisType><!--req,xs:string,"manual, DC-Iris"--></IrisType>
    <shutter>
        <!—req,xs:string,"auto, 1/1, 1/3, 1/8, 1/12, 1/15,1/25,1/30,1/50,1/60,1/100, 1/120,
        1/240, 1/480, 1/960, 1/1000, 1/2000, 1/4000, 1/8000, 1/10000,"-->
    </shutter>
    <Gain><!--req,xs:string,"auto,6dB,12dB,18dB,24dB,30dB,36dB,42dB" --></Gain>
    <Shuttermode><!--req,xs:string,"auto,manual"--></Shuttermode>
</Exposure>
```

9.6.1.5 /Images/ChannelID/WhiteBalance

/Images/ChannelID/WhiteBalance	General Resource v1.0
GET	Viewer
Description	It is used to get the whitebalance configuration of a particular channel for the device.
Query	None

Inbound Data	None
Success Return	**ResponseStatus or WhiteBlance**
PUT	Viewer
Description	It is used to set the whitebalance configuration of a particular channel for the device.
Query	None
Inbound Data	**WhiteBlance**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. When the <Mode> is "manual",<GainRed> and <GainBlue> are effective ,otherwise are not effective.	

WhiteBlance XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<WhiteBlance>
    <Mode>
        <!--req,xs,string,"auto,incandescent,warm,fluorescent,natural,manual,
        locked"-->
    </ Mode>
    <GainRed><!--dep,xs:integer,0-100--></GainRed>
    <GainBlue><!--dep,xs:integer,0-100--></GainBlue>
</WhiteBlance>
```

9.6.1.6 /Images/ChannelID/IrCutFilter

/Images/ChannelID/IrCutFilter	General Resource v1.0
GET	Viewer
Description	It is used to get the IrCutFilter configuration of a particular channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or IrCutFilter**
PUT	Viewer
Description	It is used to set the IrCutFilter configuration of a particular channel for the device.
Query	None
Inbound Data	**IrCutFilter**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

When the <Mode> is “time”,<StartTime> and <EndTime> are effective,other is not effective.When the <Mode> is “initiative” or “passivity”, <Sensitivity> and <SwitchTime> are not effective ,other is not effective.

The greater the value of the <Sensitivity> ,the more sensitive.

IrCutFilter XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<IrCutFilter>
    <Mode><!--req,xs:string,”initiative,passivity(auto),day,night,time,color,intelligent,
variableinfraredlight,variablewhitelight”--></Mode>
    <DayStartTime>
        <!--dep,xs:time, ”17:30:08”-->
    </ DayStartTime >
    <DayEndTime>
        <!--dep,xs:time, ” 17:30:08”-->
    </ DayStartTime >
    <Sensitivity><!—dep,xs:integer,0-100--></Sensitivity>
    <SwitchTime><!—dep,xs:integer,3s-15s--></SwitchTime>
    <InitiveSwitchTime><!—dep,xs:integer,3s-15s--></InitiveSwitchTime>
    <PassivitySwitchTime><!—dep,xs:integer,3s-15s--></PassivitySwitchTime>
    <ColorLastTime><!—dep,xs:integer,3s-15s--></ColorLastTime>
    <IrCutReverse><!—dep,xs:string, notsupport, true, positive--></IrCutReverse>
    <ImageMode><!—dep,xs:string, normal, facenoexposure, licenseplate --></ImageMode>
    <VariableInfraredThreshold>
        <!—dep,xs:integer,0-100-->
    </VariableInfraredThreshold>
    <VariableWhiteThreshold>
        <!—dep,xs:integer,0-100-->
    </VariableWhiteThreshold>
    <intelligentWorkmode>
        <!—dep,xs:string, constantly, flicker -->
    </intelligentWorkmode>
    <FlickerTime>
        <!—dep,xs:interger,5-180S -->
    </FlickerTime>
    <FlickerFrequency>
        <!—dep,xs:interger,0-100 -->
    </FlickerFrequency>
    <FlickerBrigthness>
        <!—dep,xs:interger,0-100 -->
    </FlickerBrigthness>
    <ColorWorkmode>
        <!—dep,xs:string, custom, auto-->
    </ColorWorkmode>
```

```
    <CustomBrightness>
        <!—dep,xs:interger,0-100 -->
    </CustomBrightness>
    <CustomStartTime>
        <!—dep,xs:datetime,20190202T020202 -->
    </CustomStartTime>
    <CustomStopTime>
        <!—dep,xs:datetime,20190202T020202 -->
    </CustomStopTime>
    <VarWhiteControlMode>
        <!—dep,xs:string, “custom”,”auto”-->
    </VarWhiteControlMode>
    <VarWhiteModeStartTime>
        <!—dep,xs:datetime,20190202T020202 -->
    </VarWhiteModeStartTime>
    <VarWhiteModeStopTime>
        <!—dep,xs:datetime,20190202T020202 -->
    </VarWhiteModeStopTime>
    <VarInfraredWorkMode>
        <!—dep,xs:string, “custom”,”auto”-->
    </VarInfraredWorkMode>
    <VarInfraredBrightness>
        <!—dep,xs:interger,0-100 -->
    </VarInfraredBrightness>
    <EnableLight><!--xs:boolean--><EnableLight>
</IrCutFilter>
```

9.7 /Pictures

/Pictures	General Resource v1.0
GET	Viewer
Description	It is used to get all picture configuration for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or PictureList**
PUT	Viewer
Description	It is used to set all picture configuration for the device.
Query	None
Inbound Data	**PictureList**
Success Return	**ResponseStatus**
Notes:	

PictureList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<PictureList>
    <Picture><!—opt-->
</PictureList>
```

9.7.1 /Pictures/ChannelID

/Pictures/ChannelID	General Resource v1.0
GET	Viewer
Description	It is used to get all picture configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Picture**
PUT	Viewer
Description	It is used to set all picture configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Picture**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Picture XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Picture>
    <ID><!—req,xs:integer--></ID>
    <OSD/><!--req-->
    <VideoLost/><!--req-->
    <Motion/><!--req-->
    <Mosaic/><!--req-->
</Picture>
```

9.7.1.1 /Pictures/ChannelID/OSD

/Pictures/ChannelID/OSD	General Resource v1.0
GET	Viewer

Description	It is used to get the OSD configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or OSD**
PUT	Viewer
Description	It is used to set the OSD configuration of a particular video input channel for the device.
Query	None
Inbound Data	**OSD**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

In setting the OSD configuration, do not let the coordinates of time and the coordinates of name are the same. In addition, the accuracy of the coordinate values of the OSD is 8, that is to say if these numbers are set 8,9,10,11,12,13,14,15 past, when the re-acquired still 8.

OSD XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<OSD>
    <DisplayTime><!--req-->
        <Enable><!--req,xs:boolean--></Enable>
        <PosX><!--req,xs:integer,0-703--></PosX>
        <PosY><!--req,xs:integer,0-575--></PosY>
    </DisplayTime>
    <DisplayName><!--req-->
        <Enable><!--req,xs:boolean--></Enable>
        <PosX><!--req,xs:integer,0-703--></PosX>
        <PosY><!--req,xs:integer,0-575--></PosY>
        <Name><!--req,xs:string,”in Unicode”--></Name>
    </DisplayName>
</OSD>
```

9.7.1.2 /Pictures/ChannelID/MultiOSD

/Pictures/ChannelID/MultiOSD	General Resource v1.0
GET	Viewer
Description	It is used to get the MultiOSD configuration of a particular video input channel for the device.
Query	None
Inbound Data	None

Success Return	ResponseStatus or MultiOSD
PUT	Viewer
Description	It is used to set the MultiOSD configuration of a particular video input channel for the device.
Query	None
Inbound Data	**MultiOSD**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

In setting the OSD configuration, do not let the coordinates of time and the coordinates of name are the same. In addition, the accuracy of the coordinate values of the OSD is 8, that is to say if these numbers are set 8,9,10,11,12,13,14,15 past, when the re-acquired still 8.

MultiOSD XML

```
<?xml version="1.0" encoding="utf-8"?>
<MultiOSD>
    <OSD><!--req-->
        <ID><!--req,xs:interger--></ID>
        <Text><!--req,xs:string--></Text>
        <x><!--req,xs:integer,0-703--></x>
        <y><!--req,xs:integer,0-575--></y>
    </OSD>
    …
    <Resolution>
        <Show><!--req,xs:boolean"true","false"--></Show>
        <x><!--req,xs:integer,0-703--></x>
        <y><!--req,xs:integer,0-575--></y>
    </Resolution>
    <Bitrate>
        <Show><!--req,xs:boolean"true","false"--></Show>
        <x><!--req,xs:integer,0-703--></x>
        <y><!--req,xs:integer,0-575--></y>
    </BitRate>
</MultiOSD>
```

9.7.1.3 /Pictures/ChannelID/MultiOSDV2

/Pictures/ChannelID/MultiOSDV2	General Resource v1.0
GET	Viewer
Description	It is used to get the MultiOSDV2 configuration of a particular video input channel for the device.
Query	None

Inbound Data	None
Success Return	**ResponseStatus or MultiLineOSD**
PUT	Viewer
Description	It is used to set the MultiOSDV2 configuration of a particular video input channel for the device.
Query	None
Inbound Data	**MultiLineOSD**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. In setting the OSD configuration, do not let the coordinates of time and the coordinates of name are the same. In addition, the accuracy of the coordinate values of the OSD is 8, that is to say if these numbers are set 8,9,10,11,12,13,14,15 past, when the re-acquired still 8.	

MultiLineOSD XML

```
<?xml version="1.0" encoding="utf-8"?>
<MultiLineOSD>
    <DisplayTime><!--req-->
        <Enable><!--req,xs:boolean--></Enable>
        <PosX><!--req,xs:integer,0-703--></PosX>
        <PosY><!--req,xs:integer,0-575--></PosY>
    </DisplayTime>
    <OSD><!--req-->
        <ID><!--req,xs:interger--></ID>
        <Text><!--req,xs:string--></Text>
        <x><!--req,xs:integer,0-703--></x>
        <y><!--req,xs:integer,0-575--></y>
    </OSD>
    …
</MultiLineOSD>
```

9.7.1.4 /Pictures/ChannelID/VideoLost

/Pictures/ChannelID/VideoLost	General Resource v1.0
GET	Viewer
Description	It is used to get the Video Lost configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or VideoLost**
PUT	Viewer

Description	It is used to set the VideoLost configuration of a particular video input channel for the device.
Query	None
Inbound Data	**VideoLost**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. For video loss, some devices are supported, some devices are not supported,For non-supported devices, though it can get some configurations, in fact, it does not make sense	

VideoLost XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<VideoLost>
    <Enable><!--req,xs:boolean--></Enable>
    <Trigger><!--req-->
    <Schedule><!--req-->
</VideoLost>
```

9.7.1.4.1 /Pictures/ChannelID/VideoLost/Trigger

/Pictures/ChannelID/VideoLost/Trigger	General Resource v1.0
GET	Viewer
Description	It is used to get the Video Lost Trigger configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Trigger**
PUT	Viewer
Description	It is used to set the Video Lost Trigger configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Trigger**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. The format of <AlarmOutMask>,<SnapshotMask>,<RecordMask> is similar to “11111111” or “00000000”, Each bit represents whether a channel is linkage or not, NO.1 means linkage, NO.0 means not linkage, for example, the string “11111111” Represents that the device is 8 channels, from left to right, represent channel 1,channel 2,channel 3…… and so on.	

For single-channel device, <AlarmOutMask>, <SnapshotMask>, <RecordMask>, <PTZActionList> is not effective .

VideoLost Trigger XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Trigger>
    <NotifyAMS>
        <Enable><!--req,xs:boolean--></Enable>
    </NotifyAMS>
    <BeepAlert>
        <Enable><!--req,xs:boolean--></Enable>
    </BeepAlert>
    <FullScreen>
        <Enable><!--req,xs:boolean--></Enable>
    </FullScreen>
    <Mail>
        <Enable><!--req,xs:boolean--></Enable>
    </Mail>
    <AlarmOut>
        <Enable><!--req,xs:boolean--></Enable>
        <AlarmOutMask>
            <!--req,xs:string,”111111111111111”-->
        </AlarmOutMask>
    </AlarmOut>
    <Snapshot>
        <Enable><!--req,xs:boolean--></Enable>
        <SnapshotMask>
            <!--req,xs:string, ”111111111111111”-->
        </SnapshotMask>
    </Snapshot>
    <Record>
        <Enable><!--req,xs:boolean--></Enable>
        <RecordMask>
            <!--req,xs:string, ”111111111111111”-->
        </RecordMask>
    </Record>
    <PTZ>
        <Enable><!--req,xs:boolean--></Enable>
        <PTZActionList>
            <PTZAction>
                <ChannelID><req,xs:integer></ChannelID>
                <ActionName><!-- req, xs:string,”none,preset,
                pattern, patrol” --></ActionName>
                <ActionNum><!--req,xs:integer,depend on <ActionName>--></ActionNum>
```

```
                </PTZAction>
            </PTZActionList>
        </PTZ>
</Trigger>
```

9.7.1.4.2 /Pictures/ChannelID/VideoLost/Schedule

/Pictures/ChannelID/VideoLost/Schedule Resource v1.0	General
GET	Viewer
Description	It is used to get the Video Lost schedule configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Schedule**
PUT	Viewer
Description	It is used to set the Video Lost schedule configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Schedule**
Success Return	ResponseStaus **ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

If the value of <AllDay> is true , indicates all time every day is valid for the motion,but the value of <TimeBlock> is invalid,if it is false,the value of <TimeBlock> is valid.

A day is divided into 48 block,which means that every half hour to a block of time,when a block of time is armed,then the block is set to 1,otherwise,the block is set to 0,for example, if all time of a day is valid,then the value of <Block_0> is "111111111111111111111111111111111111111111111111",if all time of a day is invalid,then the value of <Block_0> is "000000000000000000000000000000000000000000000000".

VideoLost Schedule XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Schedule>
    <AllDay><!--req,xs:Boolean--></AllDay>
    <TimeBlock>
        <Block_0><!req,xs:string></Block_0>
        <Block_1><!req,xs:string></Block_1>
        <Block_2><!req,xs:string></Block_2>
        <Block_3><!req,xs:string></Block_3>
```

```
            <Block_4><!req,xs:string></Block_4>
            <Block_5><!req,xs:string></Block_5>
            <Block_6><!req,xs:string></Block_6>
        <TimeBlock>
</Schedule>
```

9.7.1.5 /Pictures/ChannelID/Motion

/Pictures/ChannelID/Motion	General Resource v1.0
GET	Viewer
Description	It is used to get the motion configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Motion**
PUT	Viewer
Description	It is used to set the motion configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Motion**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. <Senstive> Some versions support for sensitivity of Highest, Higher, high,middle,low,Lower and lowest,Some versions only support for sensitivity of high,middle,low.	

Motion XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Motion>
    <Enable><!--req,xs:boolean--></Enable>
    <Senstive><!--req,xs:string,"lowest,lower,low,middle,high,higher,highest",--></Senstive>
    <Trigger><!--opt-->
    <Schedule><!--opt-->
    <MotionRegionList><!--opt-->
</Motion>
```

9.7.1.5.1 /Pictures/ChannelID/Motion/Trigger

/Pictures/ChannelID/Motion/Trigger	General Resource v1.0
GET	Viewer
Description	It is used to get the motion trigger

	configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Trigger**
PUT	Viewer
Description	It is used to set the motion trigger configuration of a particular video input channel for the device
Query	None
Inbound Data	**Trigger**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

The format of <AlarmOutMask>,<SnapshotMask>,<RecordMask> is similar to “11111111” or “00000000”, each bit represents whether a channel is linkage or not, 1 means linkage,0 means not linkage, for example, the string “11111111” represents that the device is 8 channels, from left to right, represent channel 1,channel 2,channel 3…… and so on.

For single-channel device, <AlarmOutMask>, <SnapshotMask>, <RecordMask> and<PTZActionList> are not effective .

Motion Trigger XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Trigger>
    <NotifyAMS>
        <Enable><!--req,xs:boolean--></Enable>
    </NotifyAMS>
    <BeepAlert>
        <Enable><!--req,xs:boolean--></Enable>
    </BeepAlert>
    <FullScreen>
        <Enable><!--req,xs:boolean--></Enable>
    </FullScreen>
    <Mail>
        <Enable><!--req,xs:boolean--></Enable>
    </Mail>
    <AlarmOut>
        <Enable><!--req,xs:boolean--></Enable>
        <AlarmOutMask><!--req,xs:string,” 111111111111111”--></AlarmOutMask>
    </AlarmOut>
    <Snapshot>
        <Enable><!--req,xs:boolean--></Enable>
        <SnapshotMask><!--req,xs:string,” 111111111111111”--></SnapshotMask>
    </Snapshot>
```

```
    <Record>
        <Enable><!--req,xs:boolean--></Enable>
        <RecordMask><!--req,xs:string," 111111111111111"--></RecordMask>
    </Record>
    <PTZ>
        <Enable><!--req,xs:boolean--></Enable>
        <ptzChannelID> <!--req, xs:string --></ptzChannelID>
        <ActionName><!-- req, xs:string,"none,preset, pattern, patrol" --></ActionName>
        <ActionNum><!--req,xs:integer,depend on <ActionName>--></ActionNum>
    </PTZ>
</Trigger>
```

9.7.1.5.2 /Pictures/ChannelID/Motion/Regions

/Pictures/ChannelID/Motion/Regions	General Resource v1.0
GET	Viewer
Description	It is used to get the motion regions configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Regions**
PUT	Viewer
Description	It is used to set the motion regions configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Regions**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. Use CGI or Use IE to set the motion detection area, these two approaches are independently of each other.	

Motion Regions XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<MotionRegionList>
    <Region><!--opt-->
        <ID><!--req,xs:integer--></ID>
        <TopLeftX><!--req,xs:integer,0-703--></TopLeftX>
        <TopLeftY><!--req,xs:integer,0-575--></TopLeftY>
        <BottomRightX><!--req,xs:integer,0-703--></BottomRightX>
```

```
            <BottomRightY><!--req,xs:integer,0-575--></BottomRightY>
        </Region>
</MotionRegionList>
```

9.7.1.5.3 /Pictures/ChannelID/Motion/RegionsV2

/Pictures/ChannelID/Motion/RegionsV2	General Resource v1.0
GET	Viewer
Description	It is used to get the motion regions configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or MotionRegionListV2**
PUT	Viewer
Description	It is used to set the motion regions configuration of a particular video input channel for the device.
Query	None
Inbound Data	**MotionRegionListV2**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Motion RegionsV2 XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<MotionRegionListV2>
    <RegionX_0><!--xs:string,”00000000000000”--></RegionX_0>
    <RegionX_1><!--xs:string, ”00000000000000”--></RegionX_1>
    <RegionX_2><!--xs:string, ”00000000000000”--></RegionX_2>
    <RegionX_3><!--xs:string, ”00000000000000”--></RegionX_3>
    <RegionX_4><!--xs:string, ”00000000000000”--></RegionX_4>
    <RegionX_5><!--xs:string,”00000000000000”--></RegionX_5>
    <RegionX_6><!--xs:string,”00000000000000”--></RegionX_6>
    <RegionX_7><!--xs:string,”00000000000000”--></RegionX_7>
    <RegionX_8><!--xs:string,”00000000000000”--></RegionX_8>
    <RegionX_9><!--xs:string,”00000000000000”--></RegionX_9>
    <RegionX_10><!--xs:string,”00000000000000”--></RegionX_10>
    <RegionX_11><!--xs:string,”00000000000000”--></RegionX_11>
</MotionRegionListV2>
```

9.7.1.5.4 /Pictures/ChannelID/Motion/Schedule

/Pictures/ChannelID/Motion/Schedule v1.0	General Resource
GET	Viewer
Description	It is used to get the motion schedule configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Schedule**
PUT	Viewer
Description	It is used to set the motion schedule configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

If the value of <AllDay> is true , indicates all time every day is valid for the motion,but the value of <TimeBlockList> is invalid,if it is false,the value of <TimeBlockList> is valid.

A day is divided into 48 block,which means that every half hour to a block of time,when a block of time is armed,then the block is set to 1,otherwise,the block is set to 0,for example, if all time of a day is valid,then the value of <TimeBlock_0> is "111111111111111111111111111111111111111111111111",if all time of a day is invalid,then the value of <Block_0> is "000000000000000000000000000000000000000000000000".

Motion Schedule XML Block

```
<Schedule>
    <AllDay><!--req,xs:boolean--></AllDay>
    <TimeBlockList>
        <TimeBlock_0><!req,xs:string></TimeBlock_0>
        <TimeBlock_1><!req,xs:string></TimeBlock_1>
        <TimeBlock_2><!req,xs:string></TimeBlock_2>
        <TimeBlock_3><!req,xs:string></TimeBlock_3>
        <TimeBlock_4><!req,xs:string></TimeBlock_4>
        <TimeBlock_5><!req,xs:string></TimeBlock_5>
        <TimeBlock_6><!req,xs:string></TimeBlock_6>
    <TimeBlockList>
</Schedule>
```

9.7.1.5.5 /Pictures/ChannelID/Motion/Notify/URL

/Pictures/ChannelID/Motion/Notify/URL	General Resource v1.0
GET	Viewer
Description	It is used to get the motion URL configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or URL**
PUT	Viewer
Description	It is used to set the motion URL configuration of a particular video input channel for the device.
Query	None
Inbound Data	**URL**
Success Return	**ResponseStatus**
Notes:	

Motion URL XML Block

```
<URL>
    <Protocol><!--req,xs:string--></Protocol>
    <Host><!--req,xs:string></Host>
    <Port><!--req,xs:string></Port >
    <Path><!--req,xs:string></Path >
    <Description><!--req,xs:string></Description >
    <Username><!--req,xs:string></Username >
    <Password><!--req,xs:string></Password>
</URL>
```

9.7.1.6 /Pictures/ChannelID/PeopleDetect

/Pictures/ChannelID/PeopleDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of people detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or PeopleDetect**

PUT	Viewer
Description	It is used to set the configuration of people detect
Query	None
Inbound Data	**PeopleDetect**
Success Return	**ResponseStatus**
Notes: <ChannelID> start with value:1。	

PeopleDetect XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<PeopleDetect Version="1.0">
    <Enable><!--req,xs:boolean--></Enable>
    <Senstive><!--req,xs:string,”low”,”middle”,”high”--></Senstive>
    <Track><!--req,xs:boolean--></Track>
    <AIH265Plus><!--req,xs:boolean--></AIH265Plus>
    <AIFace><!--req,xs:boolean--></AIFace>
    <Trigger Version="1.0">
        <NotifyAMS Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
        <Push Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Push>
        <AlarmOut Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <AlarmOutMask><!--req,xs:integer--></ AlarmOutMask >
        </AlarmOut>
        <Snapshot Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <SnapshotMask><!--req,xs:integer--></SnapshotMask>
        </Snapshot>
        <Record Version="1.0">
```

```
                <Enable><!--req,xs:boolean--></Enable>
                <RecordMask><!--req,xs:integer--></RecordMask>
            </Record>
            <PTZ Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <PTZActionList Version="1.0">
                    <PTZAction Version="1.0">
                        <ChannelID><!--req,xs:integer--></ChannelID>
                        <ActionName><!--req,xs:string--></ActionName>
                        <ActionNum><!--req,xs:integer--></ActionNum>
                    </PTZAction>
                </PTZActionList>
            </PTZ>
        </Trigger>
        <Schedule Version="1.0">
            <AllDay><!--req,xs:boolean--></AllDay>
            <TimeBlockList Version="1.0">

<TimeBlock_0>111111111111111111111111111111111111111111111111</TimeBlock_0>

<TimeBlock_1>111111111111111111111111111111111111111111111111</TimeBlock_1>

<TimeBlock_2>111111111111111111111111111111111111111111111111</TimeBlock_2>

<TimeBlock_3>111111111111111111111111111111111111111111111111</TimeBlock_3>

<TimeBlock_4>111111111111111111111111111111111111111111111111</TimeBlock_4>

<TimeBlock_5>111111111111111111111111111111111111111111111111</TimeBlock_5>

<TimeBlock_6>111111111111111111111111111111111111111111111111</TimeBlock_6>
            </TimeBlockList>
        </Schedule>
        <PeopleDetectRegionList Version="1.0">
            <Region Version="1.0">
                <ID><!--req,xs:integer--></ID>
                <TopLeftX><!--req,xs:integer--></TopLeftX>
                <TopLeftY><!--req,xs:integer--></TopLeftY>
                <BottomRightX><!--req,xs:integer--></BottomRightX>
                <BottomRightY><!--req,xs:integer--></BottomRightY>
            </Region>
        </PeopleDetectRegionList>
</PeopleDetect>
```

9.7.1.7 /Pictures/ChannelID/PeopleDetect/Ability

/Pictures/ChannelID/PeopleDetect/Ability	General Resource v1.0
GET	Viewer
Description	It is used to get the ability of people detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or PeopleAbility**

PeopleAbility XML

```
<?xml version="1.0" encoding="utf-8"?>
<PeopleAbility>
   <isSupportPeople><!--req,xs:boolean--></isSupportPeople>
   <TrackBox>
      <IsSupportMainStream><!--req,xs:boolean--></IsSupportMainStream>
      <IsSupportSubStream><!--req,xs:boolean--></IsSupportSubStream>
      <IsSupportSecSubStream><!--req,xs:boolean--></IsSupportSecSubStream>
   </TrackBox>
</PeopleAbility>
```

9.7.1.8 /Pictures/ChannelID/FaceDetect

/Pictures/ChannelID/FaceDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the configuration of face detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or FaceDetect**
PUT	Viewer
Description	It is used to set the configuration of face detect
Query	None
Inbound Data	**FaceDetect**
Success Return	**ResponseStatus**
Notes: <ChannelID> start with value:1。	

FaceDetect XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<FaceDetect Version="1.0">
      <Enable><!--req,xs:boolean--></Enable>
      <EnableOverlay><!--req,xs:boolean--></EnableOverlay>
      <Senstive><!--req,xs:Interger--><///Senstive>
```

```
<Trigger Version="1.0">
    <NotifyAMS Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </NotifyAMS>
    <BeepAlert Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </BeepAlert>
    <FullScreen Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </FullScreen>
    <Mail Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </Mail>
    <Ftp Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </Ftp>
    <Push Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </Push>
    <AlarmOut Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
        <AlarmOutMask><!--req,xs:integer--></ AlarmOutMask >
    </AlarmOut>
    <Snapshot Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
        <SnapshotMask><!--req,xs:integer--></SnapshotMask>
    </Snapshot>
    <Record Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
        <RecordMask><!--req,xs:integer--></RecordMask>
    </Record>
    <PTZ Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
        <PTZActionList Version="1.0">
            <PTZAction Version="1.0">
                <ChannelID><!--req,xs:integer--></ChannelID>
                <ActionName><!--req,xs:string--></ActionName>
                <ActionNum><!--req,xs:integer--></ActionNum>
            </PTZAction>
        </PTZActionList>
    </PTZ>
    <FaceMask Version="1.0">
        <Enable><!--req,xs:boolean--></Enable>
    </FaceMask>
```

```
    </Trigger>
    <Schedule Version="1.0">
        <AllDay><!--req,xs:boolean--></AllDay>
        <TimeBlockList Version="1.0">

<TimeBlock_0>111111111111111111111111111111111111111111111111</TimeBlock_0>

<TimeBlock_1>111111111111111111111111111111111111111111111111</TimeBlock_1>

<TimeBlock_2>111111111111111111111111111111111111111111111111</TimeBlock_2>

<TimeBlock_3>111111111111111111111111111111111111111111111111</TimeBlock_3>

<TimeBlock_4>111111111111111111111111111111111111111111111111</TimeBlock_4>

<TimeBlock_5>111111111111111111111111111111111111111111111111</TimeBlock_5>

<TimeBlock_6>111111111111111111111111111111111111111111111111</TimeBlock_6>
        </TimeBlockList>
    </Schedule>
</FaceDetect>
```

9.7.1.9 /Pictures/ChannelID/Mosaic

/Pictures/ChannelID/Mosaic	General Resource v1.0
GET	Viewer
Description	It is used to get the mosaic video input configuration of a particular video input channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Mosaic**
PUT	Viewer
Description	It is used to set the mosaic video input configuration of a particular video input channel for the device.
Query	None
Inbound Data	**Mosaic**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.	

Mosaic XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Mosaic>
    <Enable><!--req,xs:boolean--></Enable>
    <MosaicRegionList>
      <Region><!--opt-->
        <ID><!--req,xs:integer--></ID>
        <TopLeftX><!--req,xs:integer,0-703--></TopLeftX>
        <TopLeftY><!--req,xs:integer,0-575--></TopLeftY>
        <BottomRightX><!--req,xs:integer,0-703--></BottomRightX>
        <BottomRightY><!--req,xs:integer,0-575--></BottomRightY>
      </Region>
    </MosaicRegionList>
</Mosaic>
```

9.7.1.10 /Pictures/ChannalID/LightFillingMode

/Pictures/ChannelID/LightFillingMode	General Resource v1.0
GET	Viewer
Description	It is used to get the LightFillingMode
Query	None
Inbound Data	None
Success Return	**ResponseStatus or Pictures**
PUT	Viewer
Description	It is used to set the LightFillingMode
Query	None
Inbound Data	**Pictures**
Success Return	**ResponseStatus**
Notes: **The ChannelID start from 1.**	

Pictures Get XML

```
<?xml version="1.0" encoding="utf-8"?>
<Pictures>
    <ChannelID><!—ro,req,xs:interger--></ChannelID>
    <LightFillingMode><!—ro,req,xs:string:”PowerSaving”,”Warning”,”AlwayLight”--></LightFillin
gMode >
</Pictures>
```

Pictures Put XML

```
<?xml version="1.0" encoding="utf-8"?>
<Pictures>
    <LightFillingMode><!—ro,req,xs:string:”PowerSaving”,”Warning”,”AlwayLight”--></LightFillin
```

```
gMode >
</Pictures>
```

9.7.1.11/Pictures/ChannalID/TraversePlaneDetect

/Pictures/ChannelID/TraversePlaneDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the config of traverse plane detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or TraversePlaneDetect**
PUT	Operator
Description	It is used to get the config of traverse plane detect
Query	None
Inbound Data	**TraversePlaneDetect**
Success Return	**ResponseStatus**
Notes: ChannelID start from 1. This URL only support a part of device.	

TraversePlaneDetect XML

```
<?xml version="1.0" encoding="utf-8"?>
<TraversePlaneDetect>
    <Enable><!—ro,req,xs:boolean”true”,”false”--></Enable>
    <EnableOverlay><!—ro,req,xs:boolean:”true”,”false”--></EnableOverlay>
<Trigger Version="1.0">
        <NotifyAMS Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
```

```
            <Push Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
            </Push>
            <AlarmOut Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <AlarmOutMask><!--req,xs:integer--></ AlarmOutMask >
            </AlarmOut>
            <Snapshot Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <SnapshotMask><!--req,xs:integer--></SnapshotMask>
            </Snapshot>
            <Record Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <RecordMask><!--req,xs:integer--></RecordMask>
            </Record>
            <PTZ Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <PTZActionList Version="1.0">
                    <PTZAction Version="1.0">
                        <ChannelID><!--req,xs:integer--></ChannelID>
                        <ActionName><!--req,xs:string--></ActionName>
                        <ActionNum><!--req,xs:integer--></ActionNum>
                    </PTZAction>
                </PTZActionList>
            </PTZ>
            <FaceMask Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
            </FaceMask>
        </Trigger>
        <BeepAlert>
            <Enable><!—ro,req,xs:boolean”true”,”false”--></Enable>
        </BeepAlert>
        <FullScreen>
            <Enable><!—ro,req,xs:boolean”true”,”false”--></Enable>
        </FullScreen>
</TraversePlaneDetect>
```

9.7.1.12 /Pictures/ChannalID/FieldDetect

/Pictures/ChannelID/FieldDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the config of field detect

Query	None
Inbound Data	None
Success Return	**ResponseStatus or FieldDetect**
PUT	operator
Description	It is used to set the config of field detect
Query	None
Inbound Data	**FieldDetect**
Success Return	**ResponseStatus**

Notes:

ChannelID start from 1.

This URL only support a part of device.

FieldDetect XML

```
<?xml version="1.0" encoding="utf-8"?>
<FieldDetect>
    <Enable><!—ro,req,xs:boolean”true”,”false”--></Enable>
    <EnableOverlay><!—ro,req,xs:boolean:”true”,”false”--></EnableOverlay>
    <Trigger Version="1.0">
        <NotifyAMS Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
        <Push Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Push>
        <AlarmOut Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <AlarmOutMask><!--req,xs:integer--></ AlarmOutMask >
        </AlarmOut>
        <Snapshot Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <SnapshotMask><!--req,xs:integer--></SnapshotMask>
        </Snapshot>
        <Record Version="1.0">
```

```
            <Enable><!--req,xs:boolean--></Enable>
            <RecordMask><!--req,xs:integer--></RecordMask>
        </Record>
        <PTZ Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <PTZActionList Version="1.0">
                <PTZAction Version="1.0">
                    <ChannelID><!--req,xs:integer--></ChannelID>
                    <ActionName><!--req,xs:string--></ActionName>
                    <ActionNum><!--req,xs:integer--></ActionNum>
                </PTZAction>
            </PTZActionList>
        </PTZ>
        <FaceMask Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </FaceMask>
    </Trigger>
    <Schedule>
        <AllDay><!--req,xs:boolean--></AllDay>
        <TimeBlockList>
            <TimeBlock_0><!req,xs:string></TimeBlock_0>
            <TimeBlock_1><!req,xs:string></TimeBlock_1>
            <TimeBlock_2><!req,xs:string></TimeBlock_2>
            <TimeBlock_3><!req,xs:string></TimeBlock_3>
            <TimeBlock_4><!req,xs:string></TimeBlock_4>
            <TimeBlock_5><!req,xs:string></TimeBlock_5>
            <TimeBlock_6><!req,xs:string></TimeBlock_6>
        <TimeBlockList>
    </Schedule>
    <FieldList>
        <Field>
            <Senstive><!--req,xs:Interger--></Senstive>
            <Point>
                <PointX><!--req,xs:Interger--></PointX>
                <PointY><!--req,xs:Interger--></PointY>
            </Point>
            …
        </Field>
        …
    </FieldList>
</FieldDetect>
```

9.7.1.13 /Pictures/ChannalID/PIRDetect

/Pictures/ChannelID/PIRDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the config of PIRDetect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or PIRDetect**
PUT	Operator
Description	It is used to set the config of PIRDetect
Query	None
Inbound Data	**PIRDetect**
Success Return	**ResponseStatus**

Notes:

ChannelID start from 1.

This URL only support a part of device.

FieldDetect XML

```
<?xml version="1.0" encoding="utf-8"?>
<FieldDetect>
    <Enable><!—ro,req,xs:boolean”true”,”false”--></Enable>
    <Senstive><!—ro,req,xs:string--></Senstive>
    <Trigger Version="1.0">
        <NotifyAMS Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
        <Push Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Push>
        <AlarmOut Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
            <AlarmOutMask><!--req,xs:integer--></ AlarmOutMask >
```

```
            </AlarmOut>
            <Snapshot Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <SnapshotMask><!--req,xs:integer--></SnapshotMask>
            </Snapshot>
            <Record Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <RecordMask><!--req,xs:integer--></RecordMask>
            </Record>
            <PTZ Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
                <PTZActionList Version="1.0">
                    <PTZAction Version="1.0">
                        <ChannelID><!--req,xs:integer--></ChannelID>
                        <ActionName><!--req,xs:string--></ActionName>
                        <ActionNum><!--req,xs:integer--></ActionNum>
                    </PTZAction>
                </PTZActionList>
            </PTZ>
            <FaceMask Version="1.0">
                <Enable><!--req,xs:boolean--></Enable>
            </FaceMask>
        </Trigger>
        <Schedule>
            <AllDay><!--req,xs:boolean--></AllDay>
            <TimeBlockList>
                <TimeBlock_0><!req,xs:string></TimeBlock_0>
                <TimeBlock_1><!req,xs:string></TimeBlock_1>
                <TimeBlock_2><!req,xs:string></TimeBlock_2>
                <TimeBlock_3><!req,xs:string></TimeBlock_3>
                <TimeBlock_4><!req,xs:string></TimeBlock_4>
                <TimeBlock_5><!req,xs:string></TimeBlock_5>
                <TimeBlock_6><!req,xs:string></TimeBlock_6>
            <TimeBlockList>
        </Schedule>
        <Region>
            <ID><!--req,xs:integer--></ID>
            <TopLeftX><!--req,xs:integer--></TopLeftX>
            <TopLeftY><!--req,xs:integer--></TopLeftY>
            <BottomRightX><!--req,xs:integer--></BottomRightX>
            <BottomRightY><!--req,xs:integer--></BottomRightY>
        </Region>
    </FieldDetect>
```

9.8 /PTZ

/PTZ	General Resource v1.0
Notes:	

9.8.1 /PTZ/ChannelID/Presets/Set

/PTZ/ChannelID/Presets/Set	General Resource v1.0
PUT	Viewer
Description	It is used to set a preset.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is preset number.the range of value is 0—127.

Example:

URL：http://192.168.1.46/PTZ/1/Presets/Set

Data format:

```
PUT /PTZ/1/Presets/Set HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.2 /PTZ/ChannelID/Presets/Remove

/PTZ/ChannelID/Presets/Remove	General Resource v1.0
PUT	Viewer
Description	It is used to remove a preset.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is preset number.the range of value is 0—127.

some devices is not support for this URL.

Example:

URL： http://192.168.1.46/PTZ/1/Presets/Remove

Data format:

PUT /PTZ/1/Presets/Remove HTTP/1.1\r\n

Connection:close\r\n

Host: 192.168.0.46\r\n

Content-Length:\r\n

\r\n

Param1=1

9.8.3 /PTZ/ChannelID/Presets/Goto

/PTZ/ChannelID/Presets/Goto	General Resource v1.0
PUT	Viewer
Description	It is used to goto a preset
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is preset number.the range of value is 0—127.

Example:

URL： http://192.168.1.46/PTZ/1/Presets/Goto

Data format:

PUT /PTZ/1/Presets/Goto HTTP/1.1\r\n

Connection:close\r\n

Host: 192.168.0.46\r\n

Content-Length:\r\n

\r\n

Param1=1

9.8.4 /PTZ/ChannelID/Cruise/StartCruiseMem

/PTZ/ChannelID/Cruise/StartCruiseMem	General Resource v1.0
PUT	Viewer
Description	It is used to start to record cruise point.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is Cruise number.the range of value is 0—15.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/StartCuriseMen

Data format:

```
PUT /PTZ/1/Cruise/StartCuriseMen HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.5 /PTZ/ChannelID/Cruise/StopCruiseMem

/PTZ/ChannelID/Cruise/StopCruiseMem v1.0	General Resource
PUT	Viewer
Description	It is used to stop to record cruise point.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is Cruise number.the range of value is 0—15.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/StopCuriseMen

Data format:

```
PUT /PTZ/1/Cruise/StopCuriseMen HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.6 /PTZ/ChannelID/Cruise/ClearCruise

/PTZ/ChannelID/Cruise/ClearCruise	General Resource v1.0
PUT	Viewer
Description	It is used to Clear presets in a Cruise.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is Cruise number.the range of value is 0—15.

some devices is not support for this URL.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/ClearCruise

Data format:

```
PUT /PTZ/1/Cruise/ClearCruise HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.7 /PTZ/ChannelID/Cruise/AddPresetToCruise

/PTZ/ChannelID/Cruise/AddPresetToCruise	General Resource v1.0
PUT	Viewer
Description	It is used to add a preset to cruise.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is four params:

the “Param1” is cruise number.the range of value is 0—15.

the “Param2” is preset number, the range of value is 0--127

the “Param3” is time to stay.It is in second.

the “Param4” is moved speed,.the range of value is 1-10.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/AddPresetToCruise

Data format:

```
PUT /PTZ/1/Cruise/AddPresetToCruise HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=1&Param3=1&Param4=1
```

9.8.8 /PTZ/ChannelID/Cruise/StartCruise

/PTZ/ChannelID/Cruise/StartCruise	General Resource v1.0
PUT	Viewer
Description	It is used to start cruise.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is Cruise number.the range of value is 0—15.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/StartCruise

Data format:

```
PUT /PTZ/1/Cruise/StartCruise HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.9 /PTZ/ChannelID/Cruise/StopCruise

/PTZ/ChannelID/Cruise/StopCruise	General Resource v1.0
PUT	Viewer
Description	It is used to stop cruise.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is Cruise number.the range of value is 0—15.

Example:

URL： http://192.168.1.46/PTZ/1/Cruise/StopCruise

Data format:

```
PUT /PTZ/1/Cruise/StopCruise HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.10 /PTZ/ChannelID/Track/StartTrackMem

/PTZ/ChannelID/Track/StartTrackMem	General Resource v1.0
PUT	Viewer
Description	It is used to start to record the track.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is track number.the range of value is 0—15.

Example:

URL：http://192.168.1.46/PTZ/1/Track/StartTrackMem

Data format:

```
PUT /PTZ/1/Track/StartTrackMem HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.11 /PTZ/ChannelID/Track/StopTrackMem

/PTZ/ChannelID/Track/StopTrackMem	General Resource v1.0
PUT	Viewer
Description	It is used to stop to record the track.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

Example:

URL：http://192.168.1.46/PTZ/1/Track/StopTrackMem

Data format:

```
PUT /PTZ/1/Track/StopTrackMem HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
```

Param1=1

9.8.12 /PTZ/ChannelID/Track/StartTrack

/PTZ/ ChannelID/Track/StartTrack	General Resource v1.0
PUT	Viewer
Description	It is used to start track.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

Example:

URL： http://192.168.1.46/PTZ/1/Track/StartTrack

Data format:

```
PUT /PTZ/1/Track/StartTrack HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.13 /PTZ/ChannelID/Track/StopTrack

/PTZ/ChannelID/Track/StopTrack	General Resource v1.0
PUT	Viewer
Description	It is used to stop track.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one param,the “Param1” is track number.the range of value is 0—15.

Example:

URL： http://192.168.1.46/PTZ/1/Track/StopTrack

Data format:

```
PUT /PTZ/1/Track/StopTrack HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
```

\r\n Param1=1

9.8.14 /PTZ/ChannelID/FocusFar

/PTZ/ChannelID/FocusFar	General Resource v1.0
PUT	Viewer
Description	It is used to adjust focus far.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/FocusFar

Data format:

PUT /PTZ/1/FocusFar HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3

9.8.15 /PTZ/ChannelID/FocusNear

/PTZ/ChannelID/FocusNear	General Resource v1.0
PUT	Viewer
Description	It is used to adjust focus near.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/FocusNear

Data format:

PUT /PTZ/1/FocusNear HTTP/1.1\r\n

Connection:close\r\n

Host: 192.168.0.46\r\n

Content-Length:\r\n

\r\n

Param1=1&Param2=3

9.8.16 /PTZ/ChannelID/ApertureBigger

/PTZ/ChannelID/ApertureBigger	General Resource v1.0
PUT	Viewer
Description	It is used to adjust aperture bigger.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/ApertureBigger

Data format:

PUT /PTZ/1/ApertureBigger HTTP/1.1\r\n

Connection:close\r\n

Host: 192.168.0.46\r\n

Content-Length:\r\n

\r\n

Param1=1&Param2=3

9.8.17 /PTZ/ChannelID/ApertureSmaller

/PTZ/ChannelID/ApertureSmaller	General Resource v1.0
PUT	Viewer
Description	It is used to adjust aperture smaller
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/ApertureSmaller

Data format:

```
PUT /PTZ/1/ApertureSmaller HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.18 /PTZ/ChannelID/ZoomIn

/PTZ/ChannelID/ZoomIn	General Resource v1.0
PUT	Viewer
Description	It is used to adjust zoom in.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/ZoomIn

Data format:

```
PUT /PTZ/1/ZoomIn HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.19 /PTZ/ChannelID/ZoomOut

/PTZ/ChannelID/ZoomOut	General Resource v1.0
PUT	Viewer
Description	It is used to adjust zoom out.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/ZoomOut

Data format:

```
PUT /PTZ/1/ZoomOut HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.20 /PTZ/ChannelID/TurnLeft

/PTZ/ChannelID/TurnLeft	General Resource v1.0
PUT	Viewer
Description	It is used to turn left the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/TurnLeft

Data format:

```
PUT /PTZ/1/TurnLeft HTTP/1.1\r\n
Connection:close\r\n
```

Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1&Param2=3

9.8.21 /PTZ/ChannelID/TurnRight

/PTZ/ChannelID/TurnRight	General Resource v1.0
PUT	Viewer
Description	It is used to turn right the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown: Example: URL：http://192.168.1.46/PTZ/1/TurnRight Data format: PUT /PTZ/1/TurnRight HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1&Param2=3	

9.8.22 /PTZ/ChannelID/TurnUp

/PTZ/ChannelID/TurnUp	General Resource v1.0
PUT	Viewer
Description	It is used to turn up the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the	

“Param2” is the speed, the range of the value is 1-10, the Format as shown: Example: URL：http://192.168.1.46/PTZ/1/TurnUp Data format: PUT /PTZ/1/ TurnUp HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1&Param2=3

9.8.23 /PTZ/ChannelID/TurnDown

/PTZ/ChannelID/TurnDown	General Resource v1.0
PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown: Example: URL：http://192.168.1.46/PTZ/1/TurnDown Data format: PUT /PTZ/1/TurnDown HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1&Param2=3	

9.8.24 /PTZ/ChannelID/TurnUpLeft

/PTZ/ChannelID/TurnDown	General Resource v1.0
PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**

Success Return	ResponseStatus

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/TurnUpLeft

Data format:

```
PUT /PTZ/1/TurnDown HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.25 /PTZ/ChannelID/TurnUpRight

/PTZ/ChannelID/TurnDown	General Resource v1.0
PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/TurnUpRight

Data format:

```
PUT /PTZ/1/TurnDown HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.26 /PTZ/ChannelID/TurnDownLeft

/PTZ/ChannelID/TurnDown	General Resource v1.0

PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/TurnDownLeft

Data format:

```
PUT /PTZ/1/TurnDown HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.27 /PTZ/ChannelID/TurnDownRight

/PTZ/ChannelID/TurnDown	General Resource v1.0
PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/TurnDownRight

Data format:

```
PUT /PTZ/1/TurnDown HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.28 /PTZ/ChannelID/AutoLeftRight

/PTZ/ChannelID/TurnDown	General Resource v1.0
PUT	Viewer
Description	It is used to turn down the video
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is two params. The name of the first param is “Param1”,the name of the second param is “Param2”,if the “Param1” is “1” , it is present to start,otherwise,it is present to stop, the “Param2” is the speed, the range of the value is 1-10, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/AutoLeftRight

Data format:

```
PUT /PTZ/1/TurnDown HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1&Param2=3
```

9.8.29 /PTZ/ChannelID/Auxiliary

/PTZ/ChannelID/Auxiliary	General Resource v1.0
PUT	Viewer
Description	It is used to open the auxiliary.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one params that is “Param1” ,if the “Param1” is “1” , it is present to open,otherwise,it is present to close, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/Auxiliary

Data format:

```
PUT /PTZ/1/Auxiliary HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
```

Content-Length:\r\n \r\n Param1=1

9.8.30 /PTZ/ChannelID/Light

/PTZ/ChannelID/Light	General Resource v1.0
PUT	Viewer
Description	It is used to open the light.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one params that is “Param1” ,if the “Param1” is “1” , it is present to open,otherwise,it is present to close, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/Light

Data format:

PUT /PTZ/1/Light HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1

9.8.31 /PTZ/ChannelID/RainBrush

/PTZ/ChannelID/RainBrush	General Resource v1.0
PUT	Viewer
Description	It is used to open the rain brush.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one params that is “Param1” ,if the “Param1” is “1” , it is present to open,otherwise,it is present to close, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/RainBrush

Data format: PUT /PTZ/1/RainBrush HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1

9.8.32 /PTZ/ChannelID/Fan

/PTZ/ChannelID/Fan	General Resource v1.0
PUT	Viewer
Description	It is used to open the fan.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1. There is one params that is “Param1” ,if the “Param1” is “1” , it is present to open,otherwise,it is present to close, the Format as shown: Example: URL：http://192.168.1.46/PTZ/1/Fan Data format: PUT /PTZ/1/Fan HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1	

9.8.33 /PTZ/ChannelID/Hearter

/PTZ/ChannelID/Hearter	General Resource v1.0
PUT	Viewer
Description	It is used to open the hearter.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

There is one params that is “Param1” ,if the “Param1” is “1” , it is present to open,otherwise,it is present to close, the Format as shown:

Example:

URL：http://192.168.1.46/PTZ/1/Hearter

Data format:

```
PUT /PTZ/1/Hearter HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.34 /PTZ/ChannelID/Config

/PTZ/ChannelID/Config	General Resource v1.0
GET	Viewer
Description	It is used to get PTZ configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or PTZConfigChannel**
PUT	Viewer
Description	It is used to close the hearter
Query	None
Inbound Data	**PTZConfigChannel**
Success Return	**ResponseStatus**

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

<WatchMode> indicate start watch after some second.the value of the <WatchMode> is in second.

The Value of the <PTZActionID> is 0,1….,the range of the value is depend on the value of <PTZActionType>, if the <PTZActionType> is “preset”, the range is 0-15,if the <PTZActionType> is “cruise” or “track”, the value of the <PTZActionID> is must be 0.

The value of the <Protocol> must be the one of the value of the <ProtocolName> int the <PTZProtocolList>.

PTZConfigChannel Config XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<PTZConfigChannel>
    <Enable><!--req,xs:bool--></Enable>
    <Address><!—req,xs:string--></Address>
    <Bitrate><!—req,xs:string--></Bitrate>
```

```
    <Protocol><!—req,xs:string--></Protocol>
    <PTZWatch>
        <Enable><!--req,xs:boolean--></Enable>
        <WatchMode><!--req,xs:integer,in second--></WatchMode>
        <PTZActionType><!--req,xs:string,”preset,cruise,track”--></PTZActionType>
        <PTZActionID><!--req,xs:integer--></PTZActionID>
    </PTZWatch>
    <PTZProtocolList>
        <ProtocolCount><!--req,xs:string--></ProtocolCount>
        <PTZProtocol>
            <ProtocolName><!--req,xs:string--></ProtocolName>
        </PTZProtocol>
    </PTZProtocolList>
</PTZConfigChannel>
```

9.8.35 /PTZ/ChannelID/AdvanceConfig

/PTZ/ChannelID/AdvanceConfig	General Resource v1.0
GET	Viewer
Description	Get advanced configuration information for the PTZ
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or PTZAdvance**
PUT	Viewer
Description	Set advanced configuration information for the PTZ
Query	None
Inbound Data	**PTZAdvance**
Success Return	**ResponseStatus**
Notes: <ChannelID> Starting from 1.	

PTZAdvance Config XML

```
<PTZAdvance>
    <CruiseEnable><!--xs:boolean,”true,false”--></CruiseEnable>
    <BeginPresetNumForCruise><!--xs:interger--></BeginPresetNumForCruise>
    <BeginPresetActionForCruise><!--xs:string--></BeginPresetActionForCruise>
    <EndPresetNumForCruise><!--xs:interger--></EndPresetNumForCruise>
    <EndPresetActionForCruise><!--xs:string--></EndPresetActionForCruise>
    <WatchEnable><!--xs: boolean,”true,false”--></WatchEnable>
    <BeginPresetNumForWatch><!--xs:interger--></BeginPresetNumForWatch>
    <BeginPresetActionForWatch><!--xs:string--></BeginPresetActionForWatch>
```

```
    <EndPresetNumForWatch><!--xs:interger--></EndPresetNumForWatch>
    <EndPresetActionForWatch><!--xs:string--></EndPresetActionForWatch>
    <TrackEnable><!--xs: boolean,"true,false"--></TrackEnable>
    <BeginPresetNumForTrack><!--xs:interger--></BeginPresetNumForTrack>
    <BeginPresetActionForTrack><!--xs:string--></BeginPresetActionForTrack>
    <EndPresetNumForTrack><!--xs:interger--></EndPresetNumForTrack>
    <EndPresetActionForTrack><!--xs:string--></EndPresetActionForTrack>

</PTZAdvance>
```

9.8.36 /PTZ/ChannelID/Watch/StartWatch

/PTZ/ChannelID/Watch/StartWatch	General Resource v1.0
PUT	Viewer
Description	Start Watch
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Note:

<ChannelID> Starting from 1.

Param1 is Cruise.Range is 0-15.

Sample:

URL: http://192.168.1.46/PTZ/1/Watch/StartWatch

Data:

```
PUT /PTZ/1/Watch/StartWatch HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.37 /PTZ/ChannelID/Watch/StopWatch

/PTZ/ChannelID/Watch/StopWatch	General Resource v1.0
PUT	Viewer
Description	Stop Watch
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Note:

<ChannelID> Starting from 1.

Param1 is Cruise.Range is 0-15.

Sample:

URL： http://192.168.1.46/PTZ/1/Watch/StopWatch

Data:

```
PUT /PTZ/1/Watch/StopWatch HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.38 /PTZ/ChannelID/HumanTrack/StartHumanTrack

/PTZ/ChannelID/HumanTrack/StartHumanTrack	General Resource v1.0
PUT	Operator
Description	Start Track
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Note:

<ChannelID> Starting from 1.

Param1 is Cruise.Range is 0-15.

Sample:

URL： http://192.168.1.46/PTZ/1/HumanTrack/StartHumanTrack

Data:

```
PUT /PTZ/1/HumanTrack/StartHumanTrack HTTP/1.1\r\n
Connection:close\r\n
Host: 192.168.0.46\r\n
Content-Length:\r\n
\r\n
Param1=1
```

9.8.39 /PTZ/ChannelID/HumanTrack/StopHumanTrack

/PTZ/ChannelID/HumanTrack/Stop Human**Track**	General Resource v1.0
PUT	Operator
Description	Stop track
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:
<ChannelID> Starting from 1. Param1 is Cruise.Range is 0-15. Sample: URL: http://192.168.1.46/PTZ/1/HumanTrack/StopHumanTrack Data: PUT /PTZ/1/HumanTrack/StopHumanTrack HTTP/1.1\r\n Connection:close\r\n Host: 192.168.0.46\r\n Content-Length:\r\n \r\n Param1=1

9.8.40 /PTZ/ChannelID/Status

/PTZ/ChannelID/Status	General Resource v1.0
GET	Operator
Description	It is used to get the status of PTZ
Query	None
Inbound Data	**None**
Success Return	**PTZStatus or ResponseStatus**
Notes:	

PTZStatus XML

```
<?xml version="1.0" encoding="utf-8"?>
<PTZStatus>
    <ChannelID><!—req,xs:interger--></ChannelID>
    <Status><!—req,xs:interger--></Status>
</PTZStatus>
```

9.8.41 /PTZ/Config

/PTZ/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the config of PTZ
Query	None
Inbound Data	**None**
Success Return	**PTZConfig or ResponseStatus**
PUT	Operator
Description	It is used to set the config of PTZ
Query	None
Inbound Data	**PTZConfig**

Success Return	ResponseStatus
Note:	

PTZConfig XML

```
<?xml version="1.0" encoding="utf-8"?>
<PTZConfig>
    <PTZConfigChannel>
        <Enable><!--req,xs:string--></Enable>
        <Address><!—req,xs:interger--></Address>
        <Bitrate><!—req,xs:interger--></Bitrate>
        <Protocol><!--req,xs:string--></Protocol>
        <Speed><!--opt,xs:interger--></Speed>
        <PTZProtocolList>
            <ProtocolCount><!—req,xs:interger--></ProtocolCount>
            <PTZProtocol>
                <ProtocolName><!--req,xs:string--></ProtocolName>
            </PTZProtocol>
        </PTZProtocolList>
        <PTZWatch>
            <Enable><!--req,xs:string”true”,”false”--></Enable>
            <WatchMode><!—req,xs:interger--></WatchMode>
            <PTZActionType><!--req,xs:string--></PTZActionType>
            <PTZActionID><!—req,xs:interger--></PTZActionID>
        </PTZWatch>
    </PTZConfigChannel>
</PTZConfig>
```

9.9 /TwowayAudio

/TwoWayAudio	General Resource v1.0
Notes:	

9.9.1 /TwowayAudio/Open

/TwowayAudio/Open	General Resource v1.0
PUT	Viewer
Description	It is used to open the two way audio.
Query	None
Inbound Data	TwoWayAudio
Success Return	ResponseStatus
Notes:	

```
<TwoWayAudio>
    <VoiceMode><!—req,xs:string,"G726,G711U,G711A" --></VoiceMode>
</TwoWayAudio>
```

9.9.2 /TwowayAudio/Close

/TwowayAudio/Open	General Resource v1.0
PUT	Viewer
Description	It is used to close the two way audio.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

9.9.3 /TwowayAudio/SendData

/TwowayAudio/SendData	General Resource v1.0
PUT	Viewer
Description	It is used to send audio data.
Query	None
Inbound Data	**Audio Data**
Success Return	**ResponseStatus**
Notes:	

Example:

```
PUT /TwowayAudio/sendData HTTP/1.1
...
Content-Type: audio/basic
Content-Length: xxx
\r\n
TwowayAudio Data
```

9.9.4 /TwowayAudio/ReceiveData

/TwowayAudio/ReceiveData	General Resource v1.0

GET	Viewer
Description	It is used to receive audio data.
Query	None
Inbound Data	**None**
Success Return	**Audio Data or ResponseStatus**
Notes:	

Example:

GET /TwowayAudio/receiveData HTTP/1.1

...

HTTP/1.1 200 OK

...

Content-Type: audio/basic

Content-Length: xxx

\r\n

TwowayAudio Data...

9.10 /EventsNotification

/EventsNotification	General Resource v1.0
Notes:	

9.10.1 /EventsNotification/SubscribedEventTypeList

/EventsNotification/SubscribedEventTypeList	General Resource v1.0
GET	Viewer
Description	It is used to get a particular channel subscribed event type list.
Query	None
Inbound Data	**None**
Success Return	**SubscribedEventTypeList or ResponseStatus**
PUT	Operator
Description	It is used to subscribe a event type.
Query	None
Inbound Data	**SubscribedEventTypeList**
Success Return	**ResponseStatus**
DELETE	Operator
Description	It is used to delete subscribed event type.
Query	None

Inbound Data	SubscribedEventTypeList
Success Return	ResponseStatus
Notes: You only use “PUT” method to subscribe a event type and you only set the value of the <EventType>	

EventList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<SubscribedEventTypeList>
    <SubscribedEventType>
        <ID><!--req,xs:integer--></ID>
        <EventType><!—req,xs:string， “VMD,videolost,AlarmIn” --></EventType>
        <EventTypeDescription><!—req,xs:string--></EventTypeDescription>
    </SubscribedEventType>
</SubscribedEventTypeList>
```

9.10.2 /EventsNotification/Notify

/EventsNotification/Notify(not support)	General Resource v1.0
GET	Viewer
Description	It is used to get the event notification data stream through HTTP server push.
Query	None
Inbound Data	**None**
Success Return	**EventNotify**
Notes: This function is used to get an event notification alert stream from the media device via HTTP or HTTPS. This function does not require that a client/VMS system be added as an HTTP(S) destination on the media device. Instead, the client/VMS system can call this API to initialize a stream of event information from the device. In other words, a connection is established with the device when this function is called, and stays open to constantly receive event notifications. This API uses HTTP server-push with the MIME type multipart/mixed defined in RFC 2046. <protocol> is the protocol name, i.e. “HTTP” or “HTTPS”. <channelID> is present for video and analytics events. This URL is not support.	

EventList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotify>
```

```
    <IPAddress><!—req,xs:string--></IPAddress>
    <Port><!—req,xs:integer--></Port>
    <Protocol><!—req,xs:string--></Protocol>
    <ChannelID><!—req,xs:string--></ChannelID>
    <DateTime><!—req,xs:datetime--></DateTime>
    <EventType><!—req,xs:string--></EventType>
    <EventDescription><!—req,xs:string--></EventDescription>
</EventNotify>
```

9.10.3 /EventsNotification/EventList

/EventsNotification/EventList	General Resource v1.0
GET	Viewer
Description	It is used to get the event notification data stream through HTTP server push.
Query	None
Inbound Data	**None**
Success Return	**EventList**
Notes: Firstly,You must be subscribe a event(VMD,VideoLost,AlarmIn) you want by the URL of /EventsNotification/SubscribedEventTypeList,then,you can get the current event list by the url /EventsNotification/EventList.	

EventList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<EventList>
    <Event>
        <ID><!--req,xs:string --></ID>
        <IPAddress><!—req,xs:string--></IPAddress>
        <Port><!—req,xs:integer--></Port>
        <Protocol><!—req,xs:string--></Protocol>
        <ChannelID><!—req,xs:string--></ChannelID>
        <DateTime><!—req,xs:datetime--></DateTime>
        <EventType><!—req,xs:string--></EventType>
        <EventDescription><!—req,xs:string--></EventDescription>
    </Event>
</EventList>
```

9.11 /Alarm

/Alarm	General Resource v1.0

9.11.1 /Alarm/AlarmIn

/Alarm/AlarmIn	General Resource v1.0
GET	Viewer
Description	It is used to get all alarm-in configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AlarmInList
PUT	Operator
Description	It is used to set all alarm-in configuration.
Query	None
Inbound Data	AlarmInList
Success Return	**ResponseStatus**
Notes:	

AlarmIn XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmInList>
    <AlarmIn><!-- opt -->
</AlarmInList>
```

9.11.1.1 /Alarm/AlarmIn/ChannelID

/Alarm/AlarmIn/ChannelID	General Resource v1.0
GET	Viewer
Description	It is used to get alarm-in configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AlarmIn
PUT	Operator
Description	It is used to set alarm-in configuration.
Query	None
Inbound Data	AlarmIn
Success Return	**ResponseStatus**

Notes:

The format of <AlarmOutMask>,<SnapshotMask>,<RecordMask> is similar to “11111111” or “00000000”, each bit represents whether a channel is linkage or not, 1 means linkage,0 means not linkage, for example, the string “11111111” represents that the device is 8 channels, from left to right, represent channel 1,channel 2,channel 3…… and so on.

For single-channel device, <AlarmOutMask>, <SnapshotMask>, <RecordMask>

and<PTZActionList> are not effective .

AlarmIn XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmIn>
        <ID><!--req,xs:string --></ID>
        <Name><!—req,xs:string,max length little than 128--></Name>
        <AlarmInType><!--req,xs:string,”open,close”--></AlarmInType>
        <Schedule>
           <AllDay><!--req,xs:Boolean--></AllDay>
           <TimeBlock>
                <Block_0><!req,xs:string></Block_0>
                <Block_1><!req,xs:string></Block_1>
                <Block_2><!req,xs:string></Block_2>
                <Block_3><!req,xs:string></Block_3>
                <Block_4><!req,xs:string></Block_4>
                <Block_5><!req,xs:string></Block_5>
                <Block_6><!req,xs:string></Block_6>
          <TimeBlock>
       </Schedule>
       <Trigger>
        <NotifyAMS>
              <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert>
              <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen>
              <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail>
              <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <AlarmOut>
              <Enable><!--req,xs:boolean--></Enable>
              <AlarmOutMask><!--req,xs:string,” 111111111111111”--></AlarmOutMask>
        </AlarmOut>
        <Snapshot>
              <Enable><!--req,xs:boolean--></Enable>
              <SnapshotMask><!--req,xs:string,” 111111111111111”--></SnapshotMask>
        </Snapshot>
        <Record>
           <Enable><!--req,xs:boolean--></Enable>
           <RecordMask><!--req,xs:string,” 111111111111111”--></RecordMask>
       </Record>
```

```
        <PTZ>
            <Enable><!--req,xs:boolean--></Enable>
            <ptzChannelID> <!--req, xs:string --></ptzChannelID>
            <ActionName><!-- req, xs:string,”none,preset, pattern, patrol” --></ActionName>
            <ActionNum><!--req,xs:integer,depend on <ActionName>--></ActionNum>
        </PTZ>
    </Trigger>
</AlarmIn>
```

9.11.2 /Alarm/AlarmOut

/Alarm/AlarmOut	General Resource v1.0
GET	Viewer
Description	It is used to get all alarm-out configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AlarmOutList
PUT	Operator
Description	It is used to set all alarm-out configuration.
Query	None
Inbound Data	AlarmOutList
Success Return	**ResponseStatus**
Notes:	

AlarmOutList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmOutList>
    <AlarmOut><!-- opt -->
</AlarmOutList>
```

9.11.2.1 /Alarm/AlarmOut/ChannelID

/Alarm/AlarmOut/ChannelID	General Resource v1.0
GET	Viewer
Description	It is used to get a alarm-out configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AlarmOut
PUT	Operator
Description	It is used to set a alarm-out configuration.

Query	None
Inbound Data	AlarmOut
Success Return	**ResponseStatus**
Notes: If The value of the <Delay> is “keep”, the alarm keep output.if The value of the <Delay> is number such as 5,the alarm will be keep 5 second output.	

AlarmOut XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmOut>
    <ID><!--req,xs:string --></ID>
    <Name><!—req,xs:string,max length little than 128--></Name>
    <Delay><!—req,xs:string,”keep,custom(the unit is second)”--></Delay>
</AlarmOut>
```

9.11.2.1.1 /Alarm/AlarmOut/ChannelID/ControlMode

/Alarm/AlarmOut/ChannelID/ControlMode	General Resource v1.0
GET	Viewer
Description	It is used to get the status of controlmode
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AlarmOut
PUT	Operator
Description	It is used to get the status of controlmode
Query	None
Inbound Data	AlarmOut
Success Return	**ResponseStatus**
Notes: **ChannelID start from 1.**	

AlarmOut Get XML

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmOut>
    <ChannelID><!—ro,req,xs:interger--></ChannelID>
    <ControlMode><!—ro,req,xs:string:”true”,”false”--></ ControlMode>
</AlarmOut>
```

AlarmOut Put XML

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmOut>
    <ControlMode><!—ro,req,xs:string:”true”,”false”--></ ControlMode>
</AlarmOut>
```

9.11.3 /Alarm/AMS

/Alarm/AMS	General Resource v1.0
GET	Viewer
Description	It is used to get ams configuration
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AMS
PUT	Operator
Description	It is used to set ams configuration
Query	None
Inbound Data	AMS
Success Return	**ResponseStatus**
Notes:	

AMS XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<AMS>
    <Enable><!—rw,req,xs:boolean --></Enable>
    <ServerName><!—rw,req,xs:string --></ServerName>
    <UserName><!—rw,req,xs:string --></UserName>
    <Password><!—rw,req,xs:string --></Password>
    <ServerURL><!—rw,req,xs:string --></ServerURL>
</AMS>
```

9.11.4 /Alarm/AMS/1

/Alarm/AMS/1	General Resource v1.0
GET	Viewer
Description	It is used to get ams configuration
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AMS
PUT	Operator
Description	It is used to set ams configuration
Query	None
Inbound Data	AMS
Success Return	**ResponseStatus**
Notes:	

AMS XML

```
<?xml version="1.0" encoding="utf-8"?>
<AMS>
    <Enable><!—rw,req,xs:boolean --></Enable>
    <ServerName><!—rw,req,xs:string --></ServerName>
    <UserName><!—rw,req,xs:string --></UserName>
    <Password><!—rw,req,xs:string --></Password>
    <Message><!—rw,req,xs:string --></ServerURL>
</AMS>
```

9.11.5 /Alarm/AMS/2

/Alarm/AMS/2	General Resource v1.0
GET	Viewer
Description	It is used to get ams configuration
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or AMS
PUT	Operator
Description	It is used to set ams configuration
Query	None
Inbound Data	AMS
Success Return	**ResponseStatus**
Notes:	

AMS XML

```
<?xml version="1.0" encoding="utf-8"?>
<AMS>
    <Enable><!—rw,req,xs:boolean --></Enable>
    <ServerName><!—rw,req,xs:string --></ServerName>
    <UserName><!—rw,req,xs:string --></UserName>
    <Password><!—rw,req,xs:string --></Password>
    <Message><!—rw,req,xs:string --></ServerURL>
</AMS>
```

9.11.6 /Alarm/EventCfgList

/Alarm/EventCfgList	General Resource v1.0
GET	Viewer
Description	It is used to get event configuration list
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or EventCfgList**

PUT	Operator
Description	It is used to set event configuration list
Query	None
Inbound Data	**EventCfgList**
Success Return	**ResponseStatus**
Notes:	

EventCfgList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<EventCfgList>
    <motion>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
        <Message><!—rw,req,xs:string--></Message>
    </motion>
    <http>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
        <Message><!—rw,req,xs:string--></Message>
    </http>
</EventCfgList>
```

9.11.7 /Alarm/EventCfgList/1

/Alarm/EventCfgList/1	General Resource v1.0
GET	Viewer
Description	It is used to get the alarm event config
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or EventCfgList**
PUT	Operator
Description	It is used to set the alarm event config
Query	None
Inbound Data	**EventCfgList**
Success Return	**ResponseStatus**
Notes:	

EventCfgList XML

```
<?xml version="1.0" encoding="utf-8"?>
<EventCfgList>
    <motion>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
```

```
        <Message><!—rw,req,xs:string--></Message>
    </motion>
    <http>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
        <Message><!—rw,req,xs:string--></Message>
    </http>
</EventCfgList>
```

9.11.8 /Alarm/EventCfgList/2

/Alarm/EventCfgList/2	General Resource v1.0
GET	Viewer
Description	It is used to get the alarm event config
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or EventCfgList**
PUT	Operator
Description	It is used to set the alarm event config
Query	None
Inbound Data	**EventCfgList**
Success Return	**ResponseStatus**
Notes：	

EventCfgList XML

```
<?xml version="1.0" encoding="utf-8"?>
<EventCfgList>
    <motion>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
        <Message><!—rw,req,xs:string--></Message>
    </motion>
    <http>
        <PreRecordTime><!—rw,req,xs:interger--></PreRecordTime>
        <AlarmInterval><!—rw,req,xs:interger--></AlarmInterval>
        <Message><!—rw,req,xs:string--></Message>
    </http>
</EventCfgList>
```

9.11.9 /Alarm/AlarmAction/AlarmID/LightWarning

/Alarm/AlarmAction/AlarmID/LightWarning	General Resource v1.0
PUT	Operator
Description	It is used to control IPC to alarm by light.
Query	None
Inbound Data	**LightAlarm**
Success Return	**ResponseStatus**
Notes: AlarmID start from 1.	

LightAlarm Put XML

```
<?xml version="1.0" encoding="utf-8"?>
<LightAlarm><!--req-->
    <LightWarning><!--req,xs:string,"Twinkle,Open,Close"--></LightWarning>
    <TwinkleTime><!—ro,req,xs:interger(suggest:2)--></TwinkleTime>
</LightAlarm>
```

9.11.10 /Alarm/AlarmAction/AlarmID/LightWarningByCurrentConfig

/Alarm/AlarmAction/AlarmID/LightWarningByCurrentConfi	General Resource v1.0
PUT	Viewer
Description	It is used to control IPC to alarm by light. (According to current config)
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: AlarmID start from 1.	

9.11.11 /Alarm/PreventionControl

/Alarm/PreventionControl	General Resource v1.0
GET	Viewer
描述	It is used to get the status of prevention
查询参数	None
输入数据	**None**

返回参数	**ResponseStatus or AlarmPreventionControl**
PUT	Operator
描述	It is ssed to get the status of prevention
查询参数	None
输入数据	**AlarmPreventionControl**
返回参数	**ResponseStatus**
Notes： **The status will recovery after restart**	

AlarmPreventionControl XML

```
<?xml version="1.0" encoding="utf-8"?>
<AlarmPreventionControl>
        <Status><!—rw,req,xs:string”Disable”,”Enable”--></Status>
</AlarmPreventionControl>
```

9.12 /Record

/Record	General Resource v1.0
GET	Viewer
Description	It is used to get record configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or Record
PUT	Operator
Description	It is used to set record configuration.
Query	None
Inbound Data	Record
Success Return	**ResponseStatus**
Notes:	

Record XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Record>
   <RecordWriteCircleType><!—rw,req,xs:string,”fullnotwrite,writecircle”--></RecordWriteCircleT
   ype>
    <RecordDeleteType><!—rw,req,xs:string,”24,48,none”--></RecordDeleteType>
    <RecordStreamType><!—rw,req,xs:string,”main,sub,hdsub”--></RecordStreamType>
    <PreRecord><!—rw,opt,xs:integer,1-120,the unit is second--></PreRecord>
    <DelayRecord><!—rw,opt,xs:integer,1-120,the unit is second--></DelayRecord>
    <RecordMode><!—rw,opt,xs:string,”stop,manual,auto”--></RecordMode>
</Record>
```

9.12.1 /Record/HttpEvent/Config

/Record/HttpEvent/Config	General Resource v1.0
GET	Viewer
Description	It is used to get http event configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or HttpEventConfig
PUT	Operator
Description	It is used to set http event configuration.
Query	None
Inbound Data	HttpEventConfig
Success Return	**ResponseStatus**
Notes:	

AlarmOut XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<HttpEventConfig>
    <Enable><!--req,xs:boolean--></Enable>
</HttpEventConfig>
```

9.12.2 /Record/HttpEvent/Call

/Record/HttpEvent/Call	General Resource v1.0
PUT	Viewer
Description	It is used to Call a record event.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

9.12.3 /Record/Format/Call

/Record/Format/Call	General Resource v1.0
PUT	Viewer
Description	It is used to Format the storage device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

9.12.4 /Record/Notify/URL

/Record/Notify/URL	General Resource v1.0
GET	Viewer
Description	It is used to get the URL of the upload server.
Query	None
Inbound Data	None
Success Return	**URL or ResponseStatus**
PUT	Viewer
Description	It is used to Set the URL of the upload server.
Query	None
Inbound Data	**URL**
Success Return	**ResponseStatus**
Notes:	

URL XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<URL>
   <Description><!--req,xs: string --></Description>
   <Protocol><!--req,xs:string,such as http,ftp,https --></Protocol>
   <Host><!--req,xs: string --></Host
   <Port><!--req,xs:interger--></Port>
   <Path><!--req,xs:string--></Path>
   <Username><!--req,xs:string--></Username>
   <Password><!--req,xs:string--></Password>
</URL>
```

9.12.5 /Record/ChannelID/RecordSchedule

/Record/Notify/URL	General Resource v1.0
GET	Viewer
Description	It is used to get the record schedule
Query	None
Inbound Data	None
Success Return	**RecordSchedule or ResponseStatus**
PUT	Viewer
Description	It is used to set the record schedule
Query	None
Inbound Data	**RecordSchedule**

Success Return	ResponseStatus

Notes:

The <ChannelID> in URI shows the channel number. The <ChannelID> is start at 1.

If the value of <AllDay> is true , indicates all time every day is valid for the motion,but the value of <TimeBlockList> is invalid,if it is false,the value of <TimeBlockList> is valid.

A day is divided into 48 block,which means that every half hour to a block of time,when the record type of a block of time is timing record ,then the block is set to 1,when the record type of the block is no record, then the block is set to 0, when the record type of the block is motion record, then the block is set to 2, when the record type of the block is alarm record, then the block is set to 3.

for example:

if all time of a day is no record,

then the value of <TimeBlock_0> is

"000000000000000000000000000000000000000000000000",

if all time of a day is timing recored ,

then the value of <Block_0> is

"111111111111111111111111111111111111111111111111",

if all time of a day is motion recored,

then the value of <Block_0> is

"222222222222222222222222222222222222222222222222".

if all time of a day is alarm recored,

then the value of <Block_0> is

"333333333333333333333333333333333333333333333333".

RecordSchedule XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<RecordSchedule>
   <StreamType><!--req,xs:string,"main,sub,hdsub" --></StreamType>
   <Schedule>
      <AllDay><!--req,xs:boolean--></AllDay>
         <TimeBlockList>
            <TimeBlock_0><!req,xs:string></TimeBlock_0>
            <TimeBlock_1><!req,xs:string></TimeBlock_1>
            <TimeBlock_2><!req,xs:string></TimeBlock_2>
            <TimeBlock_3><!req,xs:string></TimeBlock_3>
            <TimeBlock_4><!req,xs:string></TimeBlock_4>
            <TimeBlock_5><!req,xs:string></TimeBlock_5>
            <TimeBlock_6><!req,xs:string></TimeBlock_6>
         <TimeBlockList>
   </Schedule>
</RecordSchedule>
```

9.13 /Disk

/Disk	General Resource v1.0
GET	Viewer

Description	It is used to get the disk configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or DiskList
PUT	Operator
Description	It is used to set the disk configuration.
Query	None
Inbound Data	DiskList
Success Return	**ResponseStatus**
Notes: The <PictureCapacityPercent> and <RecordCapacityPercent> is the actual storage size, not the percentage.	

DiskList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<DiskList>
    <Disk>
        <DiskID><!—ro,req,xs:integer,”0,1,2….”--></DiskID>
        <TotalCapacity><!—ro,req,xs:integer,the unit is MB--></TotalCapacity>
        <AvailableCapacity><!—ro,req,xs:integer,the unit is MB--></AvailableCapacity>
        <DiskStorageType><!—ro,req,xs:string,”SD,samba”--></DiskStorageType>
        <DiskStorageAttribute><!—ro,req,xs:string,”idle,writting,reading,diskexception,diskerror,re
        adonly,readwrite,nodisk,exceptionneedformat,exceptionneedChangeDisk,formatfailedneed
        reboot” --></DiskStorageAttribute>
        <PictureCapacityPercent><!—rw,req,xs:integer--></PictureCapacityPercent>
        <RecordCapacityPercent><!—rw,req,xs:integer--></RecordCapacityPercent>
        <DiskFormatStatus><!—ro,req,xs:string,”unformatted,formatting,formatted,format
failure,umount failure,device is busy”--></DiskFormatStatus>
    </Disk>
</DiskList>
```

9.14 /Playback

9.14.1 /Playback/GetRecordInfoByMonth

/Playback/GetRecordInfoByMonth	General Resource v1.0
GET	Viewer
Description	It is used to get the RecordInfo by Month.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or RecordInfoByMonth
Notes:	

From left to right,if the bit of the <RecordInfo> is “0”,then the day has no record,otherwise ,the day has record.

Example:

http://192.168.0.180/Playback/GetRecordInfoByMonth?Year=2017&Month=2

RecordInfoByMonth XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<RecordInfoByMonth>
    <RecordInfo><!—ro,req,xs:string,”00000000000000001111111111111111”--></RecordInfo>
</RecordInfoByMonth>
```

9.14.2 /Playback/GetRecordInfoByDay

/Playback/GetRecordInfoByDay	General Resource v1.0
GET	Viewer
Description	It is used to get the disk configuration.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or RecordInfoByDay

Notes:

<Channel> is start at 0,if the value is 0xff,It is present to search all channel.

<RecordType> is invalid,defaultly search all record information.

Example:

http://192.168.0.180/Playback/GetRecordInfoByDay?Year=2017&Month=2&Day=10&Channel=0&RecordType=0

RecordInfoByDay XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<RecordInfoByDay>
    <RecordInfoList>
        <RecordInfo>
            <Channel><!—ro,req,xs:interger--></Channel>
            <RecordType><!—ro,req,xs:string,”general,manual,alarm,motion”--></RecordType>
            <StartTime><!--req,xs:datetime,” 20040503173008”--></StartTime>
          <EndTime><!--req,xs:datetime,” 20040503173008”--></EndTime>
        </RecordInfo>
    </RecordInfoList>
</RecordInfoByDay>
```

9.15 RTMP

9.15.1 /RTMP/BasicConfig

/RTMP/BasicConfig	General Resource v1.0
GET	Viewer
Description	It is used to get basic RTMP settings.
Query	None
Inbound Data	**None**
Success Return	
SET	Viewer
Description	It is used to get the .
Query	None
Inbound Data	**None**
Success Return	
the <IsSupportRetrievePushAddressFromServer> of valueis “true” means device support retrieve pushing stream URL from a specific server. when “false”, the pushing stream URL must be specific by manually. <Mode> “dynamic” or “manual”. <PushURLGeneratorServer> is Server URL, this server must be ruled by our protocol. if you want to build a server, you should contact us.	

RecordInfoByMonth XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<BasicConfig>
    <IsSupportRetrievePushAddressFromServer><!--req,xs:boolean--></IsSupportRetrievePushA
ddressFromServer>
    <Mode><!--ro,req,xs:string--></Mode>
    <PushURLGeneratorServer><!--ro,req,xs:string--></PushURLGeneratorServer>
</BasicConfig>
```

9.15.2 /RTMP/ChannelID/StreamID/Config

/RTMP/StreamID/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the .
Query	None
Inbound Data	**None**
Success Return	
SET	Viewer
Description	It is used to get the .

Query	None
Inbound Data	**None**
Success Return	

Notes:

The <ChannelID> in the URI shows the channel number, The <ChannelID> is start at 1.

The <StreamID> represents stream type, generally "1" represents the main stream, “2” means "the first sub-Stream", “3” means “the second sub-stream”, and so on.

The <Enable> field and <PushURL> are valid when <Mode> of BasicConfig equals “manual”. “true” enable device to push stream to RTMP sever, “false” disable RTMP service.

The <PushURL> field is RTMP URL for pushing stream by RTMP protocol. when device working at manual mode, this field must be fill by user.

The <PlayURL> field is valid when device working at dynamic mode. User cloud open this URL by web browser to watch stream(ChannelID/StreamID).

RecordInfoByMonth XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
    <Enable><!--rw,req,xs:boolean--></Enable>
    <PushURL><!--rw,req,xs:string--></PushURL>
    <PushURL_Base64><!--rw,req,xs:string--></PushURL_Base64>
    <Status><!--ro,req,xs:string--></Status>
    <PlayURL><!--rw,req,xs:string--></PlayURL>
    <PlayURL_Base64><!--rw,req,xs:string--></PlayURL_Base64>
    <AudioEnable><!--rw,req,xs:boolean--></AudioEnable>
</Config>
```

9.16 /RecordFileList

9.16.1 /RecordFileList/Attribute

/RecordFileList/Attribute	General Resource v1.0
GET	Viewer
Description	It is used to get the Record file list attribute.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **RecordFileListAttribute**
PUT	Operator
Description	It is used to set the record file list attribute.
Query	None
Inbound Data	**RecordFileListAttribute**
Success Return	**ResponseStatus**

Notes: **<begin_time>,<end_time>,<timezone_min_between_utc> is valid when the method of the http protocol is "PUT".**

RecordFileListAttribute XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<RecordFileListAttribute>
        <RecordFileSumCount><!—ro,req,xs:integer--></RecordFileSumCount>
        <RecordFileCountPerPage><!—ro,req,xs:integer--></RecordFileCountPerPage>
        <begin_time><!—ro,opt,xs:string,"20180507 020445"--></begin_time>
        <end_time><!—ro,opt,xs:string" 20180507 020445"--></end_time>
        <timezone_min_between_utc><!—ro,opt,xs:interger,the unit is minute -->
      </timezone_min_between_utc>
</RecordFileListAttribute>
```

9.16.2 /RecordFileList/Pages/PageNum

/RecordFileList/Pages/PageNum	General Resource v1.0
GET	Viewer
Description	It is used to get the Record list of the special page number.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **RecordFileList**
Notes: Example: http://192.168.0.180/RecordFileList/Pages/1?Pathtype=0|1**(0,the <PathName> relative to the http root directory.1,the <PathName> is a absolute path.)**	

RecordFileList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<RecordFileList>
    <RecordFile>
        <PathName><!—ro,req,xs:string--></PathName>
        <FileName><!—ro,req,xs:string--></FileName>
        <StartTime><!--req,xs:datetime" 20040503 173008"--></StartTime>
        <StopTime><!--req,xs:datetime" 20040503 173008"--></StopTime>
        <FileSize><!--req,xs:interger,the unit is byte--></FileSize>
    </RecordFile>
</RecordFileList>
```

9.17/SnapshotFileList

9.17.1 /SnapshotFileList/Attribute

/SnapshotFileList/Attribute	General Resource v1.0
GET	Viewer
Description	It is used to get the snapshot file list attribute.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **SnapshotFileListAttribute**
PUT	Operator
Description	It is used to set the snapshot file list attribute.
Query	None
Inbound Data	**SnapshotFileListAttribute**
Success Return	**ResponseStatus**
Notes: **<begin_time>,<end_time>,<timezone_min_between_utc> is valid when the method of the http protocol is "PUT".**	

SnapshotFileListAttribute XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<SnapshotFileListAttribute>
        <SnapshotFileSumCount><!—ro,req,xs:integer--></SnapshotFileSumCount>
        <SnapshotFileCountPerPage><!—ro,req,xs:integer--></SnapshotFileCountPerPage>
        <begin_time><!—ro,opt,xs:string,"20180507 020445"--></begin_time>
        <end_time><!—ro,opt,xs:string"  20180507 020445"  --></end_time>
        <timezone_min_between_utc><!—ro,opt,xs:interger,the unit is minute -->
         </timezone_min_between_utc>
</SnapshotFileListAttribute>
```

9.17.2 /SnapshotFileList/Pages/PageNum

/SnapshotFileList/Pages/PageNum	General Resource v1.0
GET	Viewer
Description	It is used to get the snapshot list of the special page number.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **SnapshotFileList**
Notes: Example:	

http://192.168.0.180/SnapshotFileList/Pages/1?Pathtype=0|1**(0,the <PathName> relative to the http root directory.1,the <PathName> is a absolute path.)**

SnapshotFileList XML Block

```
<?xml version="1.0" encoding="utf-8"?>
<SnapshotFileList>
    <SnapshotFile>
        <PathName><!—ro,req,xs:string--></PathName>
        <FileName><!—ro,req,xs:string--></FileName>
        <StartTime><!--req,xs:datetime” 20040503 173008”--></StartTime>
        <StopTime><!--req,xs:datetime” 20040503 173008”--></StopTime>
        <FileSize><!--req,xs:interger,the unit is byte--></FileSize>
    </SnapshotFile>
</SnapshotFileList>
```

9.18 /FaceGroup

9.18.1 /FaceGroup/Create

/FaceGroups/Create	General Resource v1.0
PUT	Viewer
Description	It is used to create new face group
Query	None
Inbound Data	**FaceGroup**
Success Return	**ResponseStatus or** FaceGroupResponse
Notes:	

Input XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup> <GroupName><!-- xs:string,the max length is 32--></GroupName> </FaceGroup>
Output XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup> <GroupID><!-- xs:interger --></GroupID> </FaceGroup>

9.18.2 /FaceGroup/Modify

/FaceGroups/Modify	General Resource v1.0

PUT	Viewer
Description	If is used to modify the info of face library
Query	None
Inbound Data	**FaceGroup**
Success Return	**ResponseStatus**
Notes:	

Input XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup> <GroupID><!-- xs:string,the value is unique --></GroupID> <GroupName><!-- xs:string,the max length is 32--></GroupName> </FaceGroup>

9.18.3 /FaceGroup/Delete

/FaceGroups/Delete	General Resource v1.0
PUT	Viewer
Description	It is used to delete a face library
Query	None
Inbound Data	**FaceGroup**
Success Return	**ResponseStatus**
Notes:	

Input XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup>   <GroupID><!-- xs:interger,the value is unique --></GroupID>   <DelBindFace><!-- xs:boolean--></DelBindFace> </FaceGroup>

9.18.4 /FaceGroup/QueryAll

/FaceGroups/QueryAll	General Resource v1.0
PUT	Viewer
Description	It is used to query all the face groups
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:
CreateTime：Accept and return the string of date which format is ISO 8601

Output XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroups>   <FaceGroup>   <GroupID><!-- xs:interger,the value is unique --></GroupID>   <GroupName><!-- xs:string,the max length is 32--></ GroupName >   <CreateTime><!-- xs:datetime,”20040503T173008+08" --></CreateTime>   <PersonInfoCount><!-- xs:interger --></PersonInfoCount>   <AvailableStatus><!--xs:string,”available,unavailable” --></AvailableStatus>   </FaceGroup> </FaceGroups>

9.18.5 /FaceGroups/QueryByGroupID

/FaceGroups/QueryByGroupID	General Resource v1.0
PUT	Viewer
Description	It is used to query the info of face group by groupID
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

Input XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup>   <GroupID><!-- xs:interger,the value is unique --></GroupID> </FaceGroup>

Output XML
<?xml version = "1.0" encoding = "utf-8"?> <FaceGroup>   <GroupID><!-- xs:interger,the value is unique --></GroupID>   <GroupName><!-- xs:string,the max length is 32--></ GroupName >   <CreateTime><!-- xs:datetime,”20040503T173008+08" --></CreateTime>   <PersonInfoCount><!-- xs:interger --></PersonInfoCount>   <AvailableStatus><!--xs:boolean --></AvailableStatus> </FaceGroup>

9.18.6 /FaceGroups/QueryWorkState

/FaceGroups/QueryWorkState	General Resource v1.0
PUT	Viewer
Description	It is used to query the work state of recognize module.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

Output XML

```
<?xml version = "1.0" encoding = "utf-8"?>
<FaceReco>
    <WorkState><!--xs:string,"Initializing,normal,recoverying,modelling" --></WorkState>
    <ErrorCode><!--xs:interger--></ErrorCode>
</FaceReco>
```

9.18.7 /FaceGroup/CheckPersonInfo

/FaceGroup/CheckPersonInfo	General Resource v1.0
PUT	Viewer
Description	It is used to confirm the legitimacy of the person info
Query	None
Inbound Data	**PersonInfo**
Success Return	**ResponseStatus**
Notes: **Use the "#" to depart provice(**state**) and city(block).** **The field "PersonID" will be "0" while adding person info ,more than 0 while modifying person info.**	

Input XML

```
<?xml version = "1.0" encoding = "utf-8"?>
<PersonInfo>
    <PersonID><!--xs:interger --></PersonID>
    <Name><!-- xs:string, this max length is 32 --></Name>
    <Sex><!-- xs:string, "male,female" --></Sex>
    <CertificateType><!-- xs:string,"passport,IC"--></CertificateType>
```

```
    <CertificateID><!-- xs:string,the max length is 32--></CertificateID>
    <Country><!-- xs:string,the max length is 64 --></Country>
    <Address><!-- xs:string ,the mas length is 256--></Address>
    <Birthday><!-- xs:string,the max length is 32 --></Birthday>
    <UniqueID><!-- xs:string,the max length is 32 --></UniqueID>
</PersonInfo>
```

9.18.9 /FaceGroup/AddPersonInfoAndFaceImage

/FaceGroup/AddPersonInfoAndFaceImage	General Resource v1.0
PUT	Viewer
Description	It is used to add the person info and image info
Query	None
Inbound Data	**PersonInfo+binary data of image**
Success Return	**ResponseStatus**

Notes:

Inbound Data use the mode of form submission**, the last parameter must be binary data of image, the text parameter in XMLmode just use for display, the format od data submission keep submission.**

Use the "#" to depart provice(state**) and city(block).**

The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record

If the return code is 0x100F, the reponse xml will carry the "PersonID" and "UniqueID" field:

The response xml like this:

```
<ResponseStatus>
    <requestURL><!-- xs:string--></requestURL>
    <statusCode><!-- xs:interger,the value is unique--></statusCode>
    <PersonID><!-- xs:interger,the value is unique--></PersonID>
    <UniqueID><!-- xs:string--></UniqueID>
</ResponseStatus>
```

Input XML

```
----------------------------7e3172a51238
Content-Disposition: form-data; name="IgnoreError"

false
----------------------------7e3172a51238
Content-Disposition: form-data; name="Name"

ceshi
----------------------------7e3172a51238
Content-Disposition: form-data; name="Sex"
```

```
male
----------------------------7e3172a51238
Content-Disposition: form-data; name="CertificateType"

passport
----------------------------7e3172a51238
Content-Disposition: form-data; name="CertificateID"

2324
----------------------------7e3172a51238
Content-Disposition: form-data; name="GroupID"

8
----------------------------7e3172a51238
Content-Disposition: form-data; name="Country"

China
----------------------------7e3172a51238
Content-Disposition: form-data; name="Address"

Hubei#Wuhan
----------------------------7e3172a51238
Content-Disposition: form-data; name="Birthday"

20191008
----------------------------7e3172a51238
Content-Disposition: form-data; name="FaceImageID"

xxxxxxx
----------------------------7e3172a51238
Content-Disposition: form-data; name="UniqueID"

1
----------------------------7e3172a51238
Content-Disposition: form-data; name="file"; filename=""
Content-Type: image/jpeg

//Image binary Data:

----------------------------7e3172a51238--
```

Sample：

```
----------------------------7e3172a51238
Content-Disposition: form-data; name="IgnoreError"
```

```
false
-----------------------------7e3172a51238
Content-Disposition: form-data; name="Name"

ceshi
-----------------------------7e3172a51238
Content-Disposition: form-data; name="Sex"

male
-----------------------------7e3172a51238
Content-Disposition: form-data; name="CertificateType"

passport
-----------------------------7e3172a51238
Content-Disposition: form-data; name="CertificateID"

2324
-----------------------------7e3172a51238
Content-Disposition: form-data; name="GroupID"

8
-----------------------------7e3172a51238
Content-Disposition: form-data; name="Country"

China
-----------------------------7e3172a51238
Content-Disposition: form-data; name="Address"

Hubei#Wuhan
-----------------------------7e3172a51238
Content-Disposition: form-data; name="Birthday"

20191008
-----------------------------7e3172a51238
Content-Disposition: form-data; name="FaceImageID"

xxxxxxx
-----------------------------7e3172a51238
Content-Disposition: form-data; name="UniqueID"

1
-----------------------------7e3172a51238
Content-Disposition: form-data; name="file"; filename=""
```

```
Content-Type: image/jpeg

//Image binary Data:

----------------------------7e3172a51238--
```

9.18.10 /FaceGroup/DeletePersonList

/FaceGroup/DeletePersonList	General Resource v1.0
PUT	Viewer
Description	It is used to delete person info
Query	None
Inbound Data	**PersonInfoList**
Success Return	**ResponseStatus**
Notes: **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**	

Input XML

```
<PersonInfoList>
    <PersonInfo>
        <GroupID><!-- xs:interger,the value is unique --></GroupID>
        <PersonID><!-- xs:interger,the value is unique --></PersonID>
        <IsDeletePicture><!-- xs:boolean --></IsDeletePicture>
        <UniqueID><!-- xs:string,the max length is 32 --></UniqueID>
        <ForceDel><!-- xs:boolean --></ForceDel>//Completely delete
    </PersonInfo>
</PersonInfoList>
```

9.18.11 /FaceGroup/UnBindPersonInfo

/FaceGroup/UnBindPersonInfo	General Resource v1.0
PUT	Viewer
Description	It is used to unbind the person and face group
Query	None
Inbound Data	**PersonInfoList**
Success Return	**ResponseStatus**
Notes: **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**	

Input XML
<PersonInfoList>   <PersonInfo>   <DstGroupID><!-- xs:interger,the value is unique --></DstGroupID>   <PersonID><!-- xs:interger,the value is unique --></PersonID>   <UniqueID><!-- xs:string,the max length is 32 --></UniqueID>   </PersonInfo> </PersonInfoList>

9.18.12 /FaceGroup/BindPersonInfo

/FaceGroup/BindPersonInfo	General Resource v1.0
PUT	Viewer
Description	It is use to bind the person and face group
Query	None
Inbound Data	PersonInfoList
Success Return	**ResponseStatus**
Notes: **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**	

Input XML
<PersonInfoList>   <PersonInfo>   <SrcGroupID><!-- xs:interger,the value is unique --></SrcGroupID>   <DstGroupID><!-- xs:interger,the value is unique --></DstGroupID>   <PersonID><!-- xs:interger,the value is unique --></PersonID>   <UniqueID><!-- xs:string,the max length is 32 --></UniqueID>   </PersonInfo> </PersonInfoList>

9.18.13 /FaceGroup/ModifyPersonInfo

/FaceGroup/ModifyPersonInfo	General Resource v1.0
PUT	Viewer
Description	It is used to modify the person info.
Query	None
Inbound Data	**PersonInfo**
Success Return	**ResponseStatus**

Notes: **Use the "#" to depart provice(state) and city(block).** **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**

Input XML
<?xml version = "1.0" encoding = "utf-8"?> <PersonInfo> <PersonID><!—ro,xs:interger --></PersonID> <Name><!-- xs:string, this max length is 32 --></Name> <Sex><!-- xs:string, "male,female" --></Sex> <CertificateType><!--xs:string,"passport,IC,other"--></CertificateType> <CertificateID><!-- xs:string,the max length is 32--></CertificateID> <Country><!-- xs:string,the max length is 64 --></Country> <Address><!-- xs:string ,the mas length is 256--></Address> <Birthday><!-- xs:string,the max length is 32,"20190610T020202" --></Birthday> <UniqueID><!-- xs:string,the max length is 32 --></UniqueID> </PersonInfo>

9.18.14 /FaceGroup/UpdateFaceImageInfoByPerson

/FaceGroup/UpdateFaceImageInfoByPerson	General Resource v1.0
PUT	Viewer
Description	It is used to update the face image of person(they are binded.)
Query	None
Inbound Data	PersonFaceImageInfo + binary data of image
Success Return	**ResponseStatus**
Notes: **Inbound Data use the mode of form submission, the last parameter must be binary data of image, the text parameter in XMLmode just use for display, the format od data submission keep submission.** **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**	

Input XML
----------7e4200f1ab117a Content-Disposition: form-data; name="Name" Tom ----------7e4200f1ab117a Content-Disposition: form-data; name="Sex"

```
male
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="CertificateType"

IC
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="CertificateID"

0
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="PersonID"

16
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Country"

China
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Address"

Hubei#Wuhan
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Birthday"

20191008
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="FaceImageID"

1
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="file"; filename="53bff6693d82afbfeaff26f3f83068c5.jpg"
Content-Type: image/jpeg

xxxxxxx
----------------------------7e3172a51238
Content-Disposition: form-data; name="UniqueID"

//Image Binary Data:

----------------------------7e4200f1ab117a--
```

9.18.15 /FaceGroup/UpdatePersonInfoAndFaceImage

/FaceGroup/UpdatePersonInfoAndFaceImage	General Resource v1.0
PUT	Viewer
Description	It is used to update person info and face image (they are binded.)
Query	None
Inbound Data	**PersonInfo+ImageBinaryData**
Success Return	**ResponseStatus**

Notes:

Inbound Data use the mode of form submission**, the last parameter must be binary data of image, the text parameter in XMLmode just use for display, the format od data submission keep submission.**

Use the "#" to depart provice(state**) and city(block).**

The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record

Input XML

```
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Name"

Tom
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Sex"

male
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="CertificateType"

IC
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="CertificateID"

0
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="PersonID"

16
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Country"

China
----------------------------7e4200f1ab117a
```

```
Content-Disposition: form-data; name="Address"

Hubei#Wuhan
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="Birthday"

20191008
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="FaceImageID"

xxxxxxx
----------------------------7e3172a51238
Content-Disposition: form-data; name="UniqueID"

1
----------------------------7e4200f1ab117a
Content-Disposition: form-data; name="file"; filename="53bff6693d82afbfeaff26f3f83068c5.jpg"
Content-Type: image/jpeg

//Image Binary Data:

----------------------------7e4200f1ab117a--
```

Sample:

```
POST /FaceGroup/UpdatePersonInfoAndFaceImage HTTP/1.1\r\n
Host: 10.10.10.133\r\n
Connection: keep-alive\r\n
Content-Length: 41489\r\n
Authorization: Basic YWRtaW46\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36\r\n
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryqxrsbhB73oobws55\r\n
Accept: */*\r\n
Origin: http://10.10.10.133\r\n
Referer: http://10.10.10.133/face_database.html\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh,en-US;q=0.9,en;q=0.8,zh-CN;q=0.7\r\n
Cookie: language0=7; IsFaceSnapshot=false; IsFaceReco=true; userInfo=YWRtaW46; strem=1; commandPort=6060; rtspPort=554; isSupportFaceReco=TRUE; plugType=MJPEG\r\n
\r\n
----------------------------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="Name"\r\n
\r\n
Tom\r\n
----------------------------7e4200f1ab117a\r\n
```

Content-Disposition: form-data; name="Sex"\r\n
\r\n
male\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="CertificateType"\r\n
\r\n
IC\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="CertificateID"\r\n
\r\n
0\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="PersonID"\r\n
\r\n
16\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="Country"\r\n
\r\n
China\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="Address"\r\n
\r\n
Hubei#Wuhan\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="Birthday"\r\n
\r\n
20191008\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="FaceImageID"\r\n
\r\n
xxxxxxx\r\n
----------7e3172a51238\r\n
Content-Disposition: form-data; name="UniqueID"\r\n
\r\n
\r\n
----------7e4200f1ab117a\r\n
Content-Disposition: form-data; name="file";
filename="53bff6693d82afbfeaff26f3f83068c5.jpg"\r\n
Content-Type: image/jpeg\r\n
\r\n
//Image Binary Data:
----------7e4200f1ab117a--\r\n

9.18.16 /FaceGroup/QueryPersonInfoListCount

/FaceGroup/QueryPersonInfoListCount	General Resource v1.0
PUT	Viewer
Description	It is used to query the count of person in a group
Query	None
Inbound Data	PersonInfoFilter
Success Return	**ResponseStatus or** PersonInfoList

Notes:

The field Name, CertificateID, Country, Address can be empty, **Use the "#" to depart provice(**state**) and city(block).**

The field <Name> support fuzzy query, use "*" to replace character

Input XML

```
<PersonInfoFilter>
    <GroupID><!-- xs:interger,the value is unique, -1 is all --></GroupID>
    <Name><!-- xs:string --></Name>
    <Sex><!-- xs:string, "none,male,female" --></Sex>
    <CertificateType><!-- xs:string,",none,passport,IC"--></CertificateType>
    <CertificateID><!-- xs:string,the max length is 32--></CertificateID>
    <Country><!-- xs:string,the max length is 64 --></Country>
    <Address><!-- xs:string ,the mas length is 256--></Address>
    <StartBirthday>
      <!-- xs:string,the max length is 32," "20190610T020202"" -->
    </StartBirthday>
    <StopBirthday>
      <!-- xs:string,the max length is 32,"20190610T020202" -->
    </StopBirthday>
</PersonInfoFilter>
```

Output XML

```
<PersonInfoList>
    <PersonCount><!-- xs:interger --></PersonCount>
<PersonInfoList>
```

9.18.17 /FaceGroup/QueryPersonInfoList

/FaceGroup/QueryPersonInfoList	General Resource v1.0
PUT	Viewer

Description	It is used to query all the person info of a group
Query	None
Inbound Data	PersonInfoFilter
Success Return	**ResponseStatus or** PersonInfoList

Notes:

The field Name, CertificateID, Country, Address can be empty, **Use the "#" to depart provice(**state**) and city(block).**

The field <Name> support fuzzy query, use "*" to replace character

Input XML

```
<PersonInfoFilter>
    <GroupID><!-- xs:interger,the value is unique, -1 is all --></GroupID>
    <Name><!-- xs:string --></Name>
    <Sex><!-- xs:string, "none,male,female" --></Sex>
    <CertificateType><!-- xs:string,",none,passport,IC"--></CertificateType>
    <CertificateID><!-- xs:string,the max length is 32--></CertificateID>
    <Country><!-- xs:string,the max length is 64 --></Country>
    <Address><!-- xs:string ,the mas length is 256--></Address>
    <StartBirthday><!-- xs:string,the max length is 32 --></StartBirthday>
    <StopBirthday><!-- xs:string,the max length is 32 --></StopBirthday>
    <Pagesize><!--xs:integer --></Pagesize>
    <Pagenum><!--xs:integer --></Pagenum>
</PersonInfoFilter>
```

Output XML

```
<PersonInfoList>
    <PersonInfo>
        <GroupID><!-- xs:interger,the value is unique --></GroupID>
        <PersonID><!-- xs:interger,the value is unique --></PersonID>
        <Name><!-- xs:string, this max length is 32 --></Name>
        <Sex><!-- xs:string, "male,female" --></Sex>
        <CertificateType><!-- xs:string,"passport,IC"--></CertificateType>
        <CertificateID><!-- xs:string,the max length is 32--></CertificateID>
        <Country><!-- xs:string,the max length is 64 --></Country>
        <Address><!-- xs:string ,the mas length is 256--></Address>
        <Birthday><!-- xs:string,the max length is 32 --></Birthday>
        <FaceImageID><!-- xs:string,"1,2,3" --></FaceImageID>
    </PersonInfo>
<PersonInfoList>
```

9.18.18 /FaceGroup/QueryPersonImageInfo

/FaceGroup/QueryPersonImageInfo	General Resource v1.0
PUT	Viewer
Description	It is used to query the image info which bind to person by person ID.
Query	None
Inbound Data	PersonFaceImageInfo
Success Return	**ResponseStatus or image info**
Notes: **The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must be matched to the same record**	

Input XML

```
<?xml version = "1.0" encoding = "utf-8"?>
<PersonFaceImageInfo>
    <PersonID><!-- xs:interger,the value is unique --></PersonID>
    <PictureID><!-- xs:interger,"0,1,2" --></ PictureID>
    <UniqueID><!-- xs:string,the max length is 32 --></UniqueID>
</PersonFaceImageInfo>
```

9.18.19 /FaceGroup/QueryPersonInfoByPersonID

/FaceGroup/QueryPersonInfoByPersonID	General Resource v1.0
PUT	Viewer
Description	It is used to query person info by person ID
Query	None
Inbound Data	PersonInfoFilter
Success Return	**ResponseStatus or** PersonInfoList
Notes:	

Input XML

```
<PersonInfoParam>
    <GroupID><!-- xs:interger,the value is unique --></GroupID>
    <PersonID><!-- xs:interger,the value is unique --></PersonID>
</PersonInfoParam>
```

Output XML

```
<PersonInfo>
    <GroupID><!-- xs:interger,the value is unique --></GroupID>
    <PersonID><!-- xs:interger,the value is unique --></PersonID>
```

```
        <Name><!-- xs:string, this max length is 32 --></Name>
        <Sex><!-- xs:string, "male,female" --></Sex>
        <CertificateType><!-- xs:string,"passport,IC"--></CertificateType>
        <CertificateID><!-- xs:string,the max length is 32--></CertificateID>
        <Country><!-- xs:string,the max length is 64 --></Country>
        <Address><!-- xs:string ,the mas length is 256--></Address>
        <Birthday><!-- xs:string,the max length is 32 --></Birthday>
        <FaceImageID><!-- xs:string,"1,2,3" --></FaceImageID>
</PersonInfo>
```

9.18.20 /FaceGroup/ImagePersonMap

/FaceGroup/ ImagePersonMap	General Resource v1.0
GET	Viewer
Description	It is user to query the person and image info that corresponded.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus** or **ImagePersonMapList**
PUT	Operator
Description	It is user to set the person and image info that corresponded.
Query	None
Inbound Data	**ImagePersonMapList**
Success Return	**ResponseStatus**
Notes:	

ImagePersionMapList XML

```
<?xml version="1.0" encoding="utf-8"?>
<ImagePersonMapList>
    <ImagePersonMap>
        <ItemIndex><!—ro,req,xs:string--></ItemIndex>
        <PersonName><!—ro,req,xs:string--></PersonName>
        <ImageName><!--ro,req,xs:string--></ImageName>
        <Remark><!--ro,req,xs:boolean--></Remark>
    </ImagePersonMap>
</ImagePersonMapList>
```

9.18.21 /FaceGroup/RemoveImagePersonMap

/FaceGroup/RemoveImagePersonMap	General Resource v1.0
PUT	Operator

Description	It is user to remove the person and image info that corresponded.
Query	None
Inbound Data	**ImagePersonMapList**
Success Return	**ResponseStatus**
Notes:	

ImagePersionMapList XML

```
<?xml version="1.0" encoding="utf-8"?>
<ImagePersonMapList>
    <ImagePersonMap>
        <ItemIndex><!—ro,req,xs:string--></ItemIndex>
    </ImagePersonMap>
</ImagePersonMapList>
```

9.18.22 /FaceGroup/ChannelID/FaceBrightness

/FaceGroup/ChannelID/FaceBrightness	General Resource v1.0
GET	Viewer
Description	Get face brightness
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus or FaceGroup**
SET	Viewer
Description	Set face brightness
Query	None
Inbound Data	**FaceGroup**
Success Return	**ResponseStatus**
Notes: The PersonID cannot be passed when the field UniqueID is passed, and if two parameters are passed, the interface verifies that the two parameters must match the same record	

FaceGroup Get XML

```
<?xml version="1.0" encoding="utf-8"?>
<FaceGroup>
    <ChannelID><!—ro,req,xs:interger --></ChannelID>
    <FaceBrightness><!—ro,req,xs:interger:(1-100,default:50)--></FaceBrightness>
</FaceGroup>
```

FaceGroup Put XML

```
<?xml version="1.0" encoding="utf-8"?>
<FaceGroup>
    <FaceBrightness><!—ro,req,xs:interger:(1-100,default:50)--></FaceBrightness>
</FaceGroup>
```

9.19 /FaceReco

9.19.1 /FaceReco/ChannelID/RecoRuleList

/FaceReco/ChannelID/RecoRuleList	General Resource v1.0
GET	Viewer
Description	It is used to get the rule list of recognization.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or RecoRuleList**
Notes: **Each channel support max 4 rule.**	

Output XML

```
<?xml version="1.0" encoding="utf-8"?>
<RecoRuleList>
    <RecoRule>
        <Enable><!--req,xs:boolean--></Enable>
      <RecognitionRule>
          <!--xs:string,"Comparison pass,Comparison fail" -->
      </RecognitionRule>
      <RecoThresholdScheduleList>
          <RecoThresholdSchedule>
            <StartTime><!-- xs:datetime,”020301”--></StartTime>
            <StopTime><!-- xs:datetime,” 020301”--></StopTime>
            <RecoThreshold><!-- xs:interger,0-100--></RecoThreshold>
        </RecoThresholdSchedule>
      </RecoThresholdScheduleList>
      <CompareLimit><!-- xs:string,default is 1,the max is 5--></CompareLimit>
      <FaceGroupList>
          <FaceGroup>
              <Enable><!--xs:boolean --></Enable>
              <GroupID><!--xs:integer,the value is unique--></GroupID>
              <GroupName><!--xs:string,the max length is 32 --></GroupName>
          </FaceGroup>
      </FaceGroupList>
      <Schedule>
            <AllDay><!--req,xs:boolean--></AllDay>
            <TimeBlockList>
                <TimeBlock_0>
          <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
                </TimeBlock_0>
```

```
            <TimeBlock_1>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </TimeBlock_1>
            <TimeBlock_2>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </TimeBlock_2>
            <TimeBlock_3>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </TimeBlock_3>
            <TimeBlock_4>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </TimeBlock_4>
            <TimeBlock_5>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </TimeBlock_5>
            <TimeBlock_6>
        <--!req,xs:string,”1000000000000000000000100000000000000000000000001”>
            </Block_6>
      </TimeBlockList>
    </Schedule>
    <Trigger>
        <NotifyAMS>
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert>
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen>
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail>
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
        <Push Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Push>
        <AlarmOut>
            <Enable><!--req,xs:boolean--></Enable>
            <AlarmOutMask><!--req,xs:string,” 111111111111111”--></AlarmOutMask>
        </AlarmOut>
        <Snapshot>
```

```
                <Enable><!--req,xs:boolean--></Enable>
                <SnapshotMask><!--req,xs:string," 111111111111111"--></SnapshotMask>
            </Snapshot>
            <Record>
                <Enable><!--req,xs:boolean--></Enable>
                <RecordMask><!--req,xs:string," 111111111111111"--></RecordMask>
            </Record>
            <PTZ>
                <Enable><!--req,xs:boolean--></Enable>
                <PTZActionList>
                    <PTZAction Version="1.0">
                        <ptzChannelID><!--req, xs:string --></ptzChannelID>
                        <ActionName>
                            <!-- req, xs:string,"none,preset, pattern, patrol" -->
                        </ActionName>
                        <ActionNum><!--req,xs:integer,depend on <ActionName>--></ActionNum>
                    </PTZAction>
                </PTZActionList>
            </PTZ>
        </Trigger>
</RecoRule>
</RecoRuleList>
```

9.19.2 /FaceReco/ChannelID/RecoRuleList

/FaceReco/ChannelID/RecoRuleList	General Resource v1.0
PUT	Viewer
Description	It is used to set the rules of recognization.
Query	None
Inbound Data	RecoRuleList
Success Return	**ResponseStatus**
Notes:	

Input XML

```
<?xml version="1.0" encoding="utf-8"?>
<RecoRuleList>
    <RecoRule>
        <Enable><!--req,xs:boolean--></Enable>
        <RecognitionRule>
            <!--xs:string,"Comparison pass,Comparison fail" -->
        </RecognitionRule>
        <RecoThresholdScheduleList>
```

```
        <RecoThresholdSchedule>
          <StartTime><!-- xs:datetime,”020301”--></StartTime>
          <StopTime><!-- xs:datetime,” 020301”--></StopTime>
          <RecoThreshold><!-- xs:interger,0-100--></RecoThreshold>
      </RecoThresholdSchedule>
    </RecoThresholdScheduleList>
    <CompareLimit><!-- xs:string,default is 1,the max is 5--></CompareLimit>
    <FaceGroupList>
        <FaceGroup>
            <GroupID><!--xs:integer,the value is unique--></GroupID>
            <GroupName><!--xs:string,the max length is 32 --></GroupName>
        </FaceGroup>
    </FaceGroupList>
    <Schedule>
          <AllDay><!--req,xs:boolean--></AllDay>
          <TimeBlockList>
              <TimeBlock_0>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_0>
              <TimeBlock_1>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_1>
              <TimeBlock_2>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_2>
              <TimeBlock_3>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_3>
              <TimeBlock_4>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_4>
              <TimeBlock_5>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </TimeBlock_5>
              <TimeBlock_6>
        <--!req,xs:string,”100000000000000000000010000000000000000000000001”>
              </Block_6>
          </TimeBlockList>
    </Schedule>
    <Trigger>
        <NotifyAMS>
            <Enable><!--req,xs:boolean--></Enable>
        </NotifyAMS>
        <BeepAlert>
```

```
            <Enable><!--req,xs:boolean--></Enable>
        </BeepAlert>
        <FullScreen>
            <Enable><!--req,xs:boolean--></Enable>
        </FullScreen>
        <Mail>
            <Enable><!--req,xs:boolean--></Enable>
        </Mail>
        <Ftp Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Ftp>
        <Push Version="1.0">
            <Enable><!--req,xs:boolean--></Enable>
        </Push>
        <AlarmOut>
            <Enable><!--req,xs:boolean--></Enable>
            <AlarmOutMask><!--req,xs:string," 111111111111111"--></AlarmOutMask>
        </AlarmOut>
        <Snapshot>
              <Enable><!--req,xs:boolean--></Enable>
              <SnapshotMask><!--req,xs:string," 111111111111111"--></SnapshotMask>
          </Snapshot>
          <Record>
                <Enable><!--req,xs:boolean--></Enable>
                <RecordMask><!--req,xs:string," 111111111111111"--></RecordMask>
          </Record>
          <PTZ>
                <Enable><!--req,xs:boolean--></Enable>
                <PTZActionList>
                    <PTZAction Version="1.0">
                      <ptzChannelID><!--req, xs:string --></ptzChannelID>
                      <ActionName>
                          <!-- req, xs:string,"none,preset, pattern, patrol" -->
                      </ActionName>
                     <ActionNum><!--req,xs:integer,depend on <ActionName>--></ActionNum>
                   </PTZAction>
                </PTZActionList>
          </PTZ>
    </Trigger>
</RecoRule>
</RecoRuleList>
```

9.19.3 /FaceRecognition/QueryRecordCount

/FaceRecognition/QueryRecordCount	General Resource v1.0
PUT	Viewer
Description	It is used to query the count of records.
Query	None
Inbound Data	FaceRecognition
Success Return	**ResponseStatus or** RecognitionRecordList
Notes: **Time format: yyyyMMddThhmmss**	

Input XML
<FaceRecognitionFilter> <GroupID><!--xs:integer,the value is unique--></GroupID> //search in all face library when value is -1 <StartTime><!-- xs:datetime,”20040503T173008" --></StartTime> <StopTime><!-- xs:datetime,”20040503T173008" --></StopTime> <RecognitionResultType> <xs:string,"all,Comparison pass,Comparison fail,Failure of biopsy">> </RecognitionResultType> </FaceRecognitionFilter>

Output XML
<RecognitionRecord> <RecognitionRecordCount><!--xs:interger--></RecognitionRecordCount> </RecognitionRecord>

9.19.4 /FaceRecognition/QueryRecordList

/FaceRecognition/QueryRecordList	General Resource v1.0
PUT	Viewer
Description	It is used to query the history records of recognization
Query	None
Inbound Data	FaceRecognition
Success Return	**ResponseStatus or** RecognitionRecordList
Notes:	

Input XML
<FaceRecognitionFilter> <GroupID><!--xs:integer,the value is unique--></GroupID> // search in all face library

```
when value is -1
        <StartTime><!-- xs:datetime,”20040503T173008+08" --></StartTime>
        <StopTime><!-- xs:datetime,”20040503T173008+08" --></StopTime>
        <Pagesize><!--xs:integer --><Pagesize>
        <Pagenum><!--xs:integer --><Pagenum>
        < IsHasPath><!--xs:integer --></IsHasPath>
        <RecognitionResultType>
            <xs:string,"all,Comparison pass,Comparison fail,Failure of biopsy">
        </RecognitionResultType>
</FaceRecognitionFilter>
```

Output XML

```
<RecognitionRecordList>
    <RecognitionRecord>
        <LogID><!--xs:integer,the value is unique--></LogID>
        <PersonID><!-- xs:interger --></PersonID>
        <GroupName><!--xs:string,the max length is 32 --></GroupName>
        <Time><!-- xs:datetime,”20040503T173008+08" --></Time>
        <Pass><!--xs:boolean--><Pass>
        <SnapshotPath><!-- xs:string, --></SnapshotPath>
        <Similar><!-- xs:interger,0-100 --></Similar>
    </RecognitionRecord>
</RecognitionRecordList>
```

9.19.5 /FaceRecognition/DetailByRecognitionRecord

/FaceRecognition/DetailByRecognitionRecord	General Resource v1.0
PUT	Viewer
Description	It is used to query the detail of face recognization.
Query	None
Inbound Data	RecognitionInfo
Success Return	**ResponseStatus or** PersonInfo
Notes: The field <Name> support fuzzy query, use “*” to replace character	

Input XML

```
<RecognitionInfo>
    <LogID><!-- xs:integer --></LogID>
    <IsPicture><!-- xs:boolean --></IsPicture>
</RecognitionInfo>
```

Output XML
< RecognitionInfo> <Name><!-- xs:string, this max length is 32 --></Name> <Sex><!-- xs:string, "male,female" --></Sex> <CertificateType><!-- xs:string,"passport,IC"--></CertificateType> <CertificateID><!-- xs:string,the max length is 32--></CertificateID> <PersonID><!-- xs:interger --></PersonID> <Country><!-- xs:string,the max length is 64 --></Country> <Address><!-- xs:string ,the mas length is 256--></Address> <Birthday><!-- xs:string,the max length is 32 --></Birthday> <RecoResult><!--xs:boolean--></RecoResult> < RecognitionInfo>

9.19.6 /FaceRecognition/SnapshotByRecognitionRecord

/FaceRecognition/SnapshotByRecognitionRecord	General Resource v1.0
PUT	Viewer
Description	It is used to get the path of snapshot.
Query	None
Inbound Data	RecognitionInfo
Success Return	**ResponseStatus or http+binary data of image**
Notes:	

Input XML
<RecognitionInfo> <SnapshotPath><!-- xs:string --><SnapshotPath> </RecognitionInfo>

9.19.7

/FaceRecognition/RecoResultImageByRecognitionRecord

/FaceRecognition/RecoResultImageByRecognitionRecord	General Resource v1.0
PUT	Viewer
Description	It is used to get the result of recoginzation
Query	None
Inbound Data	RecognitionInfo
Success Return	**ResponseStatus or http+binary data of image**
Notes:	

Input XML
<RecognitionInfo>   <LogID><!-- xs:integer --><LogID> </RecognitionInfo>

9.19.8 /FaceRecognition/Ability

/FaceRecognition/Ability	General Resource v1.0
GET	Viewer
Description	It is used to get the ability of face recognization.
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes：	

Output XML
<RecognitionAbility>   <isSupportFaceReco><!-- xs:boolean --></isSupportFaceReco>   <FaceGroupSum><!--xs:interger--></FaceGroupSum>   <PersonPerFaceGroup><!--xs:interger--></PersonPerFaceGroup>   <PersonByAllFaceGroup><!--xs:interger--></PersonByAllFaceGroup>   <PersonImageSize><!--xs:interger,the unit is Byte,default is 256K--></PersonImageSize>   <FaceSnapshot>   <Max>   <Width><!--xs:interger,0-10000--></Width>   <Height><!--xs:interger,0-10000--></Height>   </Max>   <Min>   <Width><!--xs:interger,0-10000--></Width>   <Height><!--xs:interger,0-10000--></Height>   </Min>   <MaxResolution>   <Width><!--xs:interger--></Width>   <Height><!--xs:interger--></Height>   </MaxResolution>   </FaceSnapshot>   <TrackBox>   <IsSupportMainStream><!—xs:boolean--></IsSupportMainStream>   <IsSupportSubStream><!—xs:boolean--></IsSupportSubStream>   <IsSupportSecSubStream><!—xs:boolean--></IsSupportSecSubStream>

```
        </TrackBox>
</RecognitionAbility>
```

9.19.9 /FaceRecognition/FaceLivenessStatus

/FaceRecognition/FaceLivenessStatus	General Resource v1.0
GET	Viewer
Description	It is use to get the config of faceliveness
Query	None
Inbound Data	None
Success Return	**ResponseStatus or URL**
PUT	Viewer
Description	It is use to set the config of faceliveness
Query	None
Inbound Data	**URL**
Success Return	**ResponseStatus**
Notes:	

Motion URL XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<FaceLivenessStatus>
    <Enable><!--xs:boolean:”true”, ”false”--></Enable>
    <Score><!--xs:float:0-1--></Score>
</FaceLivenessStatus>
```

9.19.10 /FaceReco/ChannelID/BaseConfig

/FaceReco/ChannelID/BaseConfig	General Resource v1.0
GET	Viewer
Description	Get face base config
Query	None
Inbound Data	None
Success Return	**FaceBaseConfig**
PUT	Viewer
Description	Set face base config
Query	None
Inbound Data	**FaceBaseConfig**
Success Return	**ResponseStatus**
Notes:	

FaceBaseConfig XML

```
<FaceBaseConfig>
    //enable：true，false
    <EnableRecognition><--xs:boolean--></EnableRecognition>
    //gate open mode：0-not supported，1-face recognition，2-face temperature normal，3- face recognition + face temperature normal
    <OpenDoorMode><--xs:Interger--></OpenDoorMode>
</FaceBaseConfig>
```

9.20 /Face

9.20.1 /Face/ChannelID/DefaultFaceRegionInfo

/Face/ChannelID/DefaultFaceRegionInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the detect region, default max face size and default min face size
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes：	

Output XML

```
<?xml version="1.0" encoding="utf-8"?>
<DefaultFaceRegionInfo>
    <DetectRegion>
        <TopLeftX><!—dep,xs:interger--></TopLeftX>
        <TopLeftY><!—dep,xs:interger--></TopLeftY>
        <BottomRightX><!—dep,xs:interger--></BottomRightX>
        <BottomRightY><!—dep,xs:interger--></BottomRightY>
    </DetectRegion>
    <MaxFaceSize>
        <Width><!—dep,xs:interger--></Width>
        <Height><!—dep,xs:interger--></Height>
    </MaxFaceSize>
    <MinFaceSize>
        <Width><!—dep,xs:interger--></Width>
        <Height><!—dep,xs:interger--></Height>
    </MinFaceSize>
</DefaultFaceRegionInfo>
```

9.20.2 /Face/ChannelID/DetectRegion

/Face/ChannelID/DetectRegion	General Resource v1.0
GET	Viewer
Description	It is used to get the region config of face detect
Query	None
Inbound Data	None
Success Return	**FaceDetectRegion**
PUT	Viewer
Description	It is used to set the region config of face detect
Query	None
Inbound Data	**FaceDetectRegion**
Success Return	**ResponseStatus**
Notes:	

FaceDetectRegion XML

```
<FaceDetectRegion>
    <Region>
        <TopLeftX><--xs:interger,"0~10000"--></TopLeftX>
        <TopLeftY><--xs:interger,"0~10000"--></TopLeftY>
        <BottomRightX><--xs:interger,"0~10000"--></BottomRightX>
        <BottomRightY><--xs:interger,"0~10000"--></BottomRightY>
    </Region>
</FaceDetectRegion>
```

9.20.3 /Face/ChannelID/Config

/Face/ChannelID/Config	General Resource v1.0
GET	Viewer
Description	It is used to get the config of face detect
Query	None
Inbound Data	None
Success Return	**FaceConfig**
PUT	Viewer
Description	It is used to set the config of face detect
Query	None
Inbound Data	**FaceConfig**
Success Return	**ResponseStatus**

Notes:

FaceConfig XML

```
<FaceConfig>
    <SnapshotMode><--xs:string,"auto,fast,optimal,custom" --></SnapshotMode>
    <CutoutMode><--xs:string,"face_only,half_body,whole_body" --></CutoutMode>
    <ModeAuto>
        <SnapshotCount><--xs:interger,1~3--></SnapshotCount>
        <SnapshotInterval><--xs:interger,the unit is ms--></SnapshotInterval>
    </ModeAuto>
    <ModeFast>
        <SnapshotCount><--xs:interger,1~3--></SnapshotCount>
    </ModeFast>
    <ModeOptimal>
        <SnapshotCount><--xs:interger,1~3--></SnapshotCount>
    </ModeOptimal>
    <ModeCustom>
        <SnapshotCount><--xs:interger,1~3--></SnapshotCount>
        <PicQualityThreshold><--xs:interger,0-100--></PicQualityThreshold>
        <ObliqueAngle>
            <MaxValue><--xs:interger,0-180 --><MaxValue>
            <MinValue><--xs:interger,0-180 --><MinValue>
        </ObliqueAngle>
        <PitchAngle>
            <MaxValue><--xs:interger,0-180, the unit is angle--><MaxValue>
            <MinValue><--xs:interger,0-180, the unit is angle --><MinValue>
        </PitchAngle>
        <SideAngle>
            <MaxValue><--xs:interger,0-180, the unit is angle --><MaxValue>
            <MinValue><--xs:interger,0-180, the unit is angle --><MinValue>
        </SideAngle>
        <FuzzyDegree>
            <MaxValue><--xs:interger,0-100 --><MaxValue>
            <MinValue><--xs:interger,0-100 --><MinValue>
        </FuzzyDegree>
    </ModeCustom>
</FaceConfig>
```

9.20.4 /Face/ChannelID/OverlayInfo

/Face/ChannelID/OverlayInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the overlay info of face detect
Query	None
Inbound Data	None
Success Return	**FaceOverlayInfo**
PUT	Viewer
Description	It is used to set the overlay info of face detect
Query	None
Inbound Data	**FaceOverlayInfo**
Success Return	**ResponseStatus**
Notes:	

FaceConfig XML

```
<FaceOverlayInfo>
    <FaceTrackFrame>
        <EnableFaceTrackFrame><--xs:boolean--></EnableFaceTrackFrame>
        <EnableBackEndSuperposition><--xs:boolean--></ EnableBackEndSuperposition >
    </FaceTrackFrame>
    <EnableFaceStatistcs><--xs:boolean--></EnableFaceStatistcs>
    <FaceTrackFrame_SubStream>
        <EnableFaceTrackFrame><--xs:boolean--></EnableFaceTrackFrame>
        <EnableBackEndSuperposition><--xs:boolean--></ EnableBackEndSuperposition >
    </FaceTrackFrame_SubStream>
    <FaceTrackFrame_SecondSubStream>
        <EnableFaceTrackFrame><--xs:boolean--></EnableFaceTrackFrame>
        <EnableBackEndSuperposition><--xs:boolean--></ EnableBackEndSuperposition >
    </FaceTrackFrame_SecondSubStream>
</FaceOverlayInfo>
```

9.20.5 /Face/ChannelID/ClearFaceOverlayInfo

/Face/ChannelID/ClearFaceOverlayInfo	General Resource v1.0
PUT	Viewer
Description	It is used to clear the overlay info of face detect
Query	None
Inbound Data	**None**

Success Return	ResponseStatus
Notes:	

9.20.6 /Face/ChannelID/DefaultFaceRegionInfo

/Face/ChannelID/DefaultFaceRegionInfo	General Resource v1.0
GET	Viewer
Description	It is used to get the default face detect region info
Query	None
Inbound Data	None
Success Return	**DefaultFaceRegionInfo**
Notes:	

DefaultFaceRegionInfo XML

```
<DefaultFaceRegionInfo>
    <DetectRegion>
        <TopLeftX><--xs:Interger--></TopLeftX>
        <TopLeftY><--xs:Interger--></TopLeftY>
        <BottomRightX><--xs:Interger--></BottomRightX>
        <BottomRightY><--xs:Interger--></BottomRightY>
    </DetectRegion>
    <MaxFaceSize>
        <Width><--xs:Interger--></Width>
        <Height><--xs:Interger--></Height>
    </MaxFaceSize>
    <MinFaceSize>
        <Width><--xs:Interger--></Width>
        <Height><--xs:Interger--></Height>
    </MinFaceSize>
</DefaultFaceRegionInfo>
```

9.20.7 /Face/ChannelID/FixedRegionDetect

/Face/ChannelID/FixedRegionDetect	General Resource v1.0
GET	Viewer
Description	It is used to get the config of fixed region detect
Query	None
Inbound Data	None

Success Return	**ResponseStatus or FixedRegionDetectConfig**
PUT	Operator
Description	It is used to set the config of fixed region detect
Query	None
Inbound Data	**FixedRegionDetectConfig**
Success Return	**ResponseStatus**
Notes:	

FixedRegionDetectConfig XML

```
<FixedRegionDetectConfig>
    <RegionModeEnable><!—req,xs:boolean”true”,”false”--></RegionModeEnable>
    <RegionMode><!--req,xs:string”global”,”local”--></RegionMode>
    <Region>
        <TopLeftX><!—opt,xs:Interger--></TopLeftX>
        <TopLeftY><!—opt,xs:Interger--></TopLeftY>
        <BottomRightX><!—opt,xs:Interger--></BottomRightX>
        <BottomRightY><!—opt,xs:Interger--></BottomRightY>
    </Region>
</FixedRegionDetectConfig>
```

9.21 /AccessGate

9.21.1 /AccessGate/AddCard

/AccessGate/AddCard	General Resource v1.0
PUT	Operator
Description	It is used to add the access gate card
Query	None
Inbound Data	**AccessCardInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessCardInfoList XML

```
<AccessCardInfoList>
    <AccessCardInfo>
        <ID><!--req,xs:Interger--></ID>
        <CardNumber><!--req,xs:string--></CardNumber>
    </AccessCardInfo>
    …
</AccessCardInfoList>
```

9.21.2 /AccessGate/DeleteCard

/AccessGate/DeleteCard	General Resource v1.0
PUT	Operator
Description	It is used to delete the access gate card
Query	None
Inbound Data	**AccessCardInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessCardInfoList XML

```
<AccessCardInfoList>
    <AccessCardInfo>
        <ID><!--req,xs:Interger--></ID>
        <CardNumber><!--req,xs:string--></CardNumber>
    </AccessCardInfo>
    ....
</AccessCardInfoList>
```

9.21.3 /AccessGate/ModifyCard

/AccessGate/ModifyCard	General Resource v1.0
PUT	Operator
Description	It is used to modify the access gate card
Query	None
Inbound Data	**AccessCardInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessCardInfoList XML

```
<AccessCardInfoList>
    <AccessCardInfo>
        <ID><!--req,xs:Interger--></ID>
        <CardNumber><!--req,xs:string--></CardNumber>
    </AccessCardInfo>
    ...
</AccessCardInfoList>
```

9.21.4 /AccessGate/QueryAllCard

/AccessGate/QueryAllCard	General Resource v1.0
Get	Operator
Description	It is used to query all the access gate card info
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AccessCardInfoList**
Notes:	

AccessCardInfoList XML

```
<AccessCardInfoList>
    <AccessCardInfo>
        <ID><!--req,xs:Interger--></ID>
        <CardNumber><!--req,xs:string--></CardNumber>
    </AccessCardInfo>
    …
</AccessCardInfoList>
```

9.21.5 /AccessGate/AddPassword

/AccessGate/AddPassword	General Resource v1.0
PUT	Operator
Description	It is used to add the access gate password
Query	None
Inbound Data	**AccessPasswordInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessPasswordInfoList XML

```
<AccessPasswordInfoList>
    <AccessPasswordInfo>
        <ID><!--req,xs:Interger--></ID>
        <Password><!--req,xs:string--></Password>
    </AccessPasswordInfo>
    …
</AccessPasswordInfoList>
```

9.21.6 /AccessGate/DeletePassword

/AccessGate/DeletePassword	General Resource v1.0
PUT	Operator
Description	It is used to delete access gate password
Query	None
Inbound Data	**AccessPasswordInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessPasswordInfoList XML

```
<AccessPasswordInfoList>
    <AccessPasswordInfo>
        <ID><!--req,xs:Interger--></ID>
        <Password><!--req,xs:string--></Password>
    </AccessPasswordInfo>
    ....
</AccessPasswordInfoList>
```

9.21.8 /AccessGate/ModifyPassword

/AccessGate/ModifyPassword	General Resource v1.0
PUT	Operator
Description	If is used to modify access gate password
Query	None
Inbound Data	**AccessPasswordInfoList**
Success Return	**ResponseStatus**
Notes:	

AccessPasswordInfoList XML

```
<AccessPasswordInfoList>
    <AccessPasswordInfo>
        <ID><!--req,xs:Interger--></ID>
        <Password><!--req,xs:string--></Password>
    </AccessPasswordInfo>
    ...
</AccessPasswordInfoList>
```

9.21.9 /AccessGate/QueryAllPassword

/AccessGate/QueryAllPassword	General Resource v1.0
Get	Operator
Description	It is used to query all access gate password
Query	None
Inbound Data	None
Success Return	**ResponseStatus or QueryAllPassword**
Notes:	

AccessPasswordInfoList XML

```
<AccessPasswordInfoList>
    <AccessPasswordInfo>
        <ID><!--req,xs:Interger--></ID>
        <Password><!--req,xs:string--></Password>
    </AccessPasswordInfo>
    …
</AccessPasswordInfoList>
```

9.21.10 /AccessGate/QueryCardNumberByID?ID=xxx

/AccessGate/QueryCardNumberByID?ID=xxx	General Resource v1.0
Get	Viewer
Description	Query card by id
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AccessCardInfo**
Notes:	

AccessCardInfo XML

```
<AccessCardInfo>
    <ID><!--req,xs:Interger--></ID>
    <CardNumber><!--req,xs:string--></CardNumber>
</AccessCardInfo>
```

9.21.11 /AccessGate/QueryPasswordByID?ID=xxx

/AccessGate/QueryPasswordByID?ID=xxx	General Resource v1.0

Get	Viewer
Description	Query password by id
Query	None
Inbound Data	None
Success Return	**ResponseStatus or AccessPasswordInfo**
Notes:	

AccessPasswordInfo XML

```
<AccessPasswordInfo>
    <ID><!--req,xs:Interger--></ID>
    <Password><!--req,xs:string--></Password>
</AccessPasswordInfo>
```

9.22 /FaceMaskDetect

9.22.1 /FaceMaskDetect/AlarmConfig

/FaceMaskDetect/AlarmConfig	General Resource v1.0
Get	Viewer
Description	Iit is used to get the alarm config of face mask detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or FaceDetectMaskAlarm**
Put	Operator
Description	Iit is used to set the alarm config of face mask detect
Query	None
Inbound Data	FaceDetectMaskAlarm
Success Return	**ResponseStatus**
Notes:	

FaceDetectMaskAlarm XML

```
<FaceDetectMaskAlarm>
    <AlarmConfig><!--req,xs:string”NoAlarm”,”HasMaskAlarm”,”NoMaskAlarm”--></AlarmConfig
>
    <RegionModeEnable><!—req,xs:boolean”true”,”false”--></RegionModeEnable>
    <RegionMode>><!--req,xs:string”global”,”local”--></RegionMode>
    <Region>
```

```
            <TopLeftX><!—opt,xs:Interger--></TopLeftX>
            <TopLeftY><!—opt,xs:Interger--></TopLeftY>
            <BottomRightX><!—opt,xs:Interger--></BottomRightX>
            <BottomRightY><!—opt,xs:Interger--></BottomRightY>
        </Region>
</FaceDetectMaskAlarm>
```

9.23 /FaceTemperatureDetect

9.23.1 /FaceTemperatureDetect/ChannelID/FaceTemperatureDetectConfig

/FaceTemperatureDetect/ChannelID/FaceTemperatureDetectConfig	General Resource v1.0
Get	Viewer
Description	It is used to get the config of face temperature detect
Query	None
Inbound Data	None
Success Return	**ResponseStatus or FaceTemperatureDetectConfig**
Put	Operator
Description	It is used to set the config of face temperature detect
Query	None
Inbound Data	FaceTemperatureDetectConfig
Success Return	**ResponseStatus**
Notes: **ChannelID Start from 1.** **URL usage:** **/FaceTemperatureDetect/1/FaceTemperatureDetectConfig** **ChannelID in XML is no required When put**	

FaceTemperatureDetectConfig XML

```
<FaceTemperatureDetectConfig>
    <ChannelID><!—opt,xs:Interger--></ChannelID>
    <Enable><!—req,xs:boolean"true","false"--></Enable>
    <TemperatureType><!—req,xs:string"Fahrenheit","Centigrade"--></Temperature>
    <NormalTemperatureDownLimit><!—req,xs:float--></NormalTemperatureDownLimit>
```

```
    <NormalTemperatureUpLimit><!—req,xs:float--></NormalTemperatureUpLimit >
    <TemperatureReviseEnable><!—req,xs:boolean”true”,”false”--></TemperatureReviseEnable>
    <TemperatureReviseValue><!—req,xs:float--></TemperatureReviseValue>
<FaceTemperatureDetectConfig>
```

9.23.2 /FaceTemperatureDetect/ChannelID/OverlayFaceTemperatureOnOSD

/FaceTemperatureDetect/ChannelID/OverlayFaceTemperatureOnOSD General Resource 1.0	
Get	Viewer
Description	It is used to get the config about whether overlay the temperature info on OSD.
Query	None
Inbound Data	None
Success Return	**ResponseStatus or OverlayTemperatureOnOSDConfig**
Put	Operator
Description	It is used to set the config about whether overlay the temperature info on OSD.
Query	None
Inbound Data	**OverlayTemperatureOnOSDConfig**
Success Return	**ResponseStatus**
Notes: **ChannelID Start from 1.** **URL usage:** **/FaceTemperatureDetect/1/OverlayFaceTemperatureOnOSD** **ChannelID in XML is no required When put**	

OverlayTemperatureOnOSDConfig XML

```
<OverlayTemperatureOnOSDConfig>
    <ChannelID><!—opt,xs:Interger--></ChannelID>
    <Enable><!—req,xs:boolean”true”,”false”--></Enable>
</OverlayTemperatureOnOSDConfig>
```

9.23.3

/FaceTemperatureDetect/ChannelID/FaceTemperatureStatus

/FaceTemperatureDetect/ChannelID/FaceTemperatureStatus	General Resource v1.0
Get	Viewer
Description	It is used to get the current face temperature
Query	None
Inbound Data	None
Success Return	**ResponseStatus or FaceTemperatureStatus**
Notes: **ChannelID Start from 1.** **URL usage:** **/FaceTemperatureDetect/1/FaceTemperatureStatus**	

FaceTemperatureStatus XML

```
<FaceTemperatureStatus>
    <ChannelID><!—req,xs:Interger--></ChannelID>
    <Temperature><!—req,xs:float--></Temperature>
</FaceTemperatureStatus>
```

9.23.4

/FaceTemperatureDetect/ChannelID/FaceTemperatureAbnormalAlarmConfig

/FaceTemperatureDetect/ChannelID/FaceTemperatureAbnormalAlarmConfig General Resource v1.0	
Get	Viewer
Description	It is used to get the alarm config when the face temperature is abnormal
Query	None
Inbound Data	None
Success Return	**ResponseStatus or FaceTemperatureAbnormalAlarmConfig**
Put	Operator
Description	It is used to set the alarm config when the face temperature is abnormal
Query	None
Inbound Data	**FaceTemperatureAbnormalAlarmConfig**

Success Return	ResponseStatus
Notes: **ChannelID Start from 1.** **URL usage:** **/FaceTemperatureDetect/1/FaceTemperatureAbnormalAlarmConfig** **ChannelID in XML is no required When put**	

FaceTemperatureAbnormalAlarmConfig XML

```
<FaceTemperatureAbnormalAlarmConfig>
    <ChannelID><!—req,xs:Interger--></ChannnelID>
    <VoiceAlert><!—req,xs:string”Enable”,”Disable”--></VoiceAlert>
    <DisableWhiteListLinkage><!—req,xs:string”Enable”,”Disable”--></DisableWhiteListLin
kage>
    <PushToPhone><!—req,xs:string”Enable”,”Disable”--></PushToPhone>
    <AlarmOut1><!—req,xs:string”Enable”,”Disable”--></AlarmOut1>
    <AlarmOut2><!—req,xs:string”Enable”,”Disable”--></AlarmOut2>
</FaceTemperatureAbnormalAlarmConfig>
```

9.24 /Common

9.24.1 /Common/DeviceFileDownload

/Common/DeviceFileDownload	General Resource v1.0
GET	Viewer
Description	It is used to export the face library
Query	FileType,Path
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: **FileType = 11 used to export the face library**	

Output Param
The binary data of image

9.24.2 /Common/DeviceFileUpload

/Common/DeviceFileUpload	General Resource v1.0
PUT	Viewer
Description	It is used to import the face library
Query	None

Inbound Data	The binary data of face library
Success Return	ResponseStatus
Notes:	

Output Param
ResponseStatus

10 Device discovery V2

The IP Device in addition to support the upnp protocol to discovery device ,but this method can not set the network configuration overlay IP, so we support other method to discovery device and set the network configuration .this method use Muticast to transfer device information,the Muticast Address is 239.239.239.52,the port is 5566,the client only transfer a request packetage to the network on the 5566 port in the muticast method, we can get the device information but please note that this method is not over http,of course ,we alse can use this method to set the network configuration,

The following is the description for use.

```
//Detect Request
Client:
        <?xml version="1.0" encoding="UTF-8">
        <Request>
                <Head Version="1.0.0" ConfigLength="xxx" CommandID="10" />
                <CfgInfo MainCommand="100" AssistCommand="-1" DeviceMac="xx" />
        </Request>

Server:
        <?xml version="1.0" encoding="UTF-8">
        <Response>
                <Head Version="1.0.0" ConfigLength="xxx" CommandID="11" ReplyMsg="xx">
                <CfgInfo MainCommand="100" AssistCommand="-1" DeviceMac="xx" />
                <DeviceProbeConfig>
                        <DeviceType><!--ro,req,xs:string--></DeviceType>
                        <DeviceVersion><!--ro,req,xs:string--></DeviceVersion>
                        <DeviceChannelCount><!--ro,req,xs:integer--></DeviceChannelCount>
                        <DeviceEnableRegister><!--req,xs:boolean--></DeviceEnableRegister>
                        <DeviceRegisterInterval>
                                <!--req,xs:integer,"second"-->
                        </DeviceRegisterInterval>
                        <DeviceMac><!--ro,req,xs:string--></DeviceMac>
                        <DeviceIP><!--req,xs:string--></DeviceIP>
```

```
                <DeviceMask><!--req,xs:string--><DeviceMask>
                <DeviceGateway><!--req,xs:string--></DeviceGateway>
                <DeviceManagerHost><!--req,xs:string--></DeviceManagerHost>
                <DeviceDetectPort><!--req,xs:integer--></DeviceDetectPort>
                <ManagerHostPort><!--req,xs:integer--></ManagerHostPort>
                <CommandPort><!--req,xs:integer--></CommandPort>
                <MediaPort><!--req,xs:integer--></MediaPort>
                <HttpPort><!--req,xs:integer--></HttpPort>
                <RtspPort><!--req,xs:integer--></RtspPort>
                <TutkUID><!—ro,req,xs:integer--><TutkUID>
                <UlkUID><!—ro,req,xs:integer--><UlkUID>
            </DeviceProbeConfig>
        </Response>

//Detect Set Device Configuration
Client:
        <?xml version="1.0" encoding="UTF-8">
        <Request>
            <Head Version="1.0.0" ConfigLength="xxx" CommandID="10">
            <CfgInfo MainCommand="101" AssistCommand="-1" DeviceMac="xx" />
            <DeviceProbeConfig>
                <DeviceType><!--ro,req,xs:string--></DeviceType>
                <DeviceVersion><!--ro,req,xs:string--></DeviceVersion>
                <DeviceChannelCount><!--ro,req,xs:integer--></DeviceChannelCount>
                <DeviceEnableRegister><!--req,xs:boolean--></DeviceEnableRegister>
                <DeviceRegisterInterval>
                    <!--req,xs:integer,"second">
                </DeviceEnableRegInterval>
                <DeviceMac><!--ro,req,xs:string--></DeviceMac>
                <DeviceIP><!--req,xs:string--></DeviceIP>
                <DeviceNetmask><!--req,xs:string--><DeviceMask>
                <DeviceGateway><!--req,xs:string--></DeviceGateway>
                <DeviceManagerHost><!--req,xs:string--></DeviceManagerHost>
                <DetectPort><!--req,xs:integer--></DeviceDetectPort>
                <ManagerHostPort><!--req,xs:integer--></ManagerHostPort>
                <CommandPort><!--req,xs:integer--></CommandPort>
                <MediaPort><!--req,xs:integer--></MediaPort>
                <HttpPort><!--req,xs:integer--></HttpPort>
                <RtspPort><!--req,xs:integer--></RtspPort>
                <TutkUID><!—ro,req,xs:integer--><TutkUID>
                <UlkUID><!—ro,req,xs:integer--><UlkUID>
            </DeviceProbeConfig>
        </Request >
```

Server:

```
<?xml version="1.0" encoding="UTF-8">
<Response>
    <Head Version="x.x.x" ConfigLength="xxx" CommandID="11" ReplyMsg="xx" />
    <CfgInfo MainCommand="101" AssistCommand="-1" DeviceMac="xx" />
</Response>
```

//Detect Network Configuration , **need** **authentication**

Client:

```
<?xml version="1.0" encoding="UTF-8">
<Request>
    <Head Version="1.0.0" ConfigLength="xxx" CommandID="10" />
    <CfgInfo MainCommand="102" AssistCommand="-1" DeviceMac="xx" />
    <Auth UserName="xx" UserPassword="xx" />
</Request>
```

Server:

```
<?xml version="1.0" encoding="UTF-8">
<Response>
    <Head Version="1.0.0" ConfigLength="xxx" CommandID="11" ReplyMsg="xx">
    <CfgInfo MainCommand="102" AssistCommand="-1" DeviceMac="xx" />
    <DeviceProbeNetConfig>
        <DeviceType><!--ro,req,xs:string--></DeviceType>
        <DeviceVersion><!--ro,req,xs:string--></DeviceVersion>
        <DeviceChannelCount><!--ro,req,xs:integer--></DeviceChannelCount>
        <DeviceEnableRegister><!--req,xs:boolean--></DeviceEnableRegister>
        <DeviceEnableRegInterval>
            <!--req,xs:integer,second in unit-->
        </DeviceEnableRegInterval>
        <NetworkInterfaceList>
            <NetworkInterface>
                <ID><!--req,xs:string--></ID>
                <Enable><!--ro,req,xs:string,"open,close,notsupport"--></Enable>
                <DhcpEnable><!--req,xs:boolean--></DhcpEnable>
                <MAC><!--req,xs:string--></MAC>
                <IPAddress><!--req,xs:string--></IPAddress>
                <Netmask><!--req,xs:string--></Netmask>
                <Gateway><!--req,xs:string--></Gateway>
                <EnableAutoGenerateIP>
                    <!--req,xs:boolean-->
                </EnableAutoGenerateIP>
                <NetworkState>
```

```
                <!--ro,opt,"get ip success,close,getting ip,get ip failed"-->
            </NetworkState>
        </NetworkInterface>
    </NetworkInterfaceList>
    <DeviceManagerHost><!--req,xs:string--></DeviceManagerHost>
    <DeviceDetectPort><!--req,xs:integer--></DeviceDetectPort>
    <ManagerHostPort><!--req,xs:integer--></ManagerHostPort>
    <CommandPort><!--req,xs:integer--></CommandPort>
    <MediaPort><!--req,xs:integer--></MediaPort>
    <HttpPort><!--req,xs:integer--></HttpPort>
    <RtspPort><!--req,xs:integer--></RtspPort>
    <TutkUID><!—ro,req,xs:integer--><TutkUID>
    <UlkUID><!—ro,req,xs:integer--><UlkUID>
  </DeviceProbeNetConfig>
</Response>
```

//Detect Set Network Configuration ,**need** **authentication**

Client:

```
<?xml version="1.0" encoding="UTF-8">
<Request>
    <Head Version="1.0.0" ConfigLength="xxx" CommandID="10">
  <CfgInfo MainCommand="103" AssistCommand="-1" DeviceMac="xx" />
  <Auth UserName="xx" UserPassword="xx" />
  <DeviceProbeNetConfig>
    <DeviceType><!--ro,req,xs:string--></DeviceType>
    <DeviceVersion><!--ro,req,xs:string--></DeviceVersion>
    <DeviceChannelCount><!--ro,req,xs:integer--></DeviceChannelCount>
    <DeviceEnableRegister><!--req,xs:boolean--></DeviceEnableRegister>
    <DeviceEnableRegInterval>
        <!--req,xs:integer,second in unit-->
    </DeviceEnableRegInterval>
    <DeviceMac><!--ro,req,xs:string--></DeviceMac>
    <NetworkInterfaceList>
        <NetworkInterface>
            <ID><!--req,xs:string--></ID>
            <Enable><!--req,xs:string,"open,close,notsupport"--></Enable>
            <DhcpEnable><!--req,xs:boolean--></DhcpEnable>
            <MAC><!--req,xs:string--></MAC>
            <IPAddress><!--req,xs:string--></IPAddress>
            <Netmask><!--req,xs:string--></Netmask>
            <Gateway><!--req,xs:string--></Gateway>
            <EnableAutoGenerateIP>
                <!--req,xs:boolean-->
            </EnableAutoGenerateIP>
```

```
                <NetworkState>
                  <!--opt,"get ip success,close,getting ip,get ip failed"-->
                </NetworkState>
            </NetworkInterface>
        </NetworkInterfaceList>
        <DeviceManagerHost><!--req,xs:string--></DeviceManagerHost>
        <DeviceDetectPort><!--req,xs:integer--></DeviceDetectPort>
        <ManagerHostPort><!--req,xs:integer--></ManagerHostPort>
        <CommandPort><!--req,xs:integer--></CommandPort>
        <MediaPort><!--req,xs:integer--></MediaPort>
        <HttpPort><!--req,xs:integer--></HttpPort>
        <RtspPort><!--req,xs:integer--></RtspPort>
        <TutkUID><!—ro,req,xs:integer--><TutkUID>
        <UlkUID><!—ro,req,xs:integer--><UlkUID>
    </DeviceProbeNetConfig>
</Request>
```

Server:

```
<?xml version="1.0" encoding="UTF-8" />
<Response>
    <Head Version="1.0.0" ConfigLength="xxx" CommandID="11" ReplyMsg="xx" />
    <CfgInfo MainCommand="103" AssistCommand="-1" DeviceMac="xx" />
</Response>
```

Note:

<Version> is show this interface version, now, we only support "1.0.0".

<ConfigLenght> is show the struct length that you send.

<CommandID> is the Command ID,if it is the data that the client send ,the <CommandID> is 10, If it is the data that the server send,the <CommandID> is 11.

<Replymsg> is the error code that the server send, if the <ReplyMsg> is 0,then reply success,otherwise, reply failed.

<MainCommand> and <AssistCommand> is show the instruction that you execute. Different request has different instruction.

<DeviceMac> is the Device mac that you want to set.