Определение состояния публикуемого аудио/видео: заглушено/не заглушено

- Получение состояния потока при подключении к потоку

В сборке IOS SDK2.6.32добавлена возможность получать при проигрывании потока события от публикующей стороны. Таким образом можно определить, заглушено ли аудио или видео в потоке: если клиент использует функцию muteAduio() или muteVideo(), подписчикам отправляется специальное событие. Чтобы получить это событие при проигрывании потока, необходимо переопределить функцию FPWCSApi2Stream.onStrea mEvent в Objective C приложении или WCSStream.onStreamEvent в Swift приложении.

Пример для Objective C code

```
    [_remoteStream onStreamEvent:^(FPWCSApi2StreamEvent *streamEvent){
        NSLog(@"No remote stream, %@", streamEvent.type);
        if ([streamEvent.type isEqual:[FPWCSApi2Model streamEventTypeToString:
kFPWCSStreamEventTypeAudioMuted]]) {
            [_remoteControl onAudioMute:true];
        }
        if ([streamEvent.type isEqual:[FPWCSApi2Model streamEventTypeToString:
kFPWCSStreamEventTypeAudioUnmuted]]) {
            [_remoteControl onAudioMute:false];
        }
        if ([streamEvent.type isEqual:[FPWCSApi2Model streamEventTypeToString:
kFPWCSStreamEventTypeVideoMuted]]) {
            [_remoteControl onVideoMute:true];
        }
        if ([streamEvent.type isEqual:[FPWCSApi2Model streamEventTypeToString:
kFPWCSStreamEventTypeVideoUnmuted]]) {
            [_remoteControl onVideoMute:false];
        }
    }];
```

Пример для Swift code

```
            playStream?.onStreamEvent({streamEvent in
                if (streamEvent?.type == FPWCSApi2Model.streamEventType(toString: .
fpwcsStreamEventTypeAudioMuted)) {
                    self.remoteViewController?.onAudioMute(true);
                }
                if (streamEvent?.type == FPWCSApi2Model.streamEventType(toString: .
fpwcsStreamEventTypeAudioUnmuted)) {
                    self.remoteViewController?.onAudioMute(false);
                }
                if (streamEvent?.type == FPWCSApi2Model.streamEventType(toString: .
fpwcsStreamEventTypeVideoMuted)) {
                    self.remoteViewController?.onVideoMute(true);
                }
                if (streamEvent?.type == FPWCSApi2Model.streamEventType(toString: .
fpwcsStreamEventTypeVideoUnmuted)) {
                    self.remoteViewController?.onVideoMute(false);
                }
            });
```

Получение состояния потока при подключении к потоку

При подключении подписчика к потоку, состояние потока можно получить в обработчике событияStreamStatusPlaying,используя методы Stream. getAudioState() и Stream.getVideoState()

Пример для Objective C code

```
    [_remoteStream on:kFPWCSStreamStatusPlaying callback:^(FPWCSApi2Stream *stream){
        [self changeStreamStatus:stream];
        [self onStarted];
        _useLoudSpeaker.control.userInteractionEnabled = YES;
        [_remoteControl onAudioMute:[stream getAudioState].muted];
        [_remoteControl onVideoMute:[stream getVideoState].muted];
    }];
```

Пример для Swift code

```
    @IBAction func playPressed(_ sender: Any) {
        changeViewState(playButton,false)
        if (playButton.title(for: .normal) == "PLAY") {
            let options = FPWCSApi2StreamOptions()
            options.name = playName.text;
            options.display = remoteDisplay.videoView;
            options.constraints = remoteMediaConstrains;
            options.transport = tcpTransport.isOn ? kFPWCSTransport.fpwcsTransportTCP : kFPWCSTransport.
fpwcsTransportUDP;
            do {
            playStream = try session!.createStream(options)
            } catch {
                print(error);
            }
            playStream?.on(.fpwcsStreamStatusPlaying, {rStream in
                self.changeStreamStatus(rStream!)
                self.onPlaying(rStream!);
            });
            ...
        }
    }
    ...
    fileprivate func onPlaying(_ stream:FPWCSApi2Stream) {
        playButton.setTitle("STOP", for:.normal)
        changeViewState(loudSpeaker, true)
        changeViewState(playButton, true)
        self.remoteViewController!.onAudioMute(stream.getAudioState()?.muted ?? false)
        self.remoteViewController!.onVideoMute(stream.getVideoState()?.muted ?? false)
    }
```