

Video publishing and playback on mobile devices

- [Common rules](#)
- [Stream publishing and playback in Low Power Mode](#)

Common rules

Mobile browser requires user action (button click) when starting to publish or play a media stream on mobile device. Also, audio must be muted in HTML5 video tag:

```
let video = document.createElement('video');
video.muted = true;
```

When playing a stream, audio can be unmuted by user action only.

To workaround this, since WebSDK build [2.0.212](#) stream publishing or playback should be started on mobile device by the following rules:

1. If Websocket connection is already established, and a separate button is used to start publishing or playback (Publish/Play button which user clicks), no additional actions required in application code
2. If Websocket connection is establishing by button click, then publishing/playback starts by `SESSION_STATUS.ESTABLISHED` event, it is necessary to use `playFirstVideo()` function for publishing

[code](#)

```
if (Browser.isSafariWebRTC()) {
    Flashphoner.playFirstVideo(localVideo, true, PRELOADER_URL).then(function() {
        publishStream();
    });
    return;
}
```

and playback

[code](#)

```
if (Flashphoner.getMediaProviders()[0] === "WSPlayer") {
    ...
} else if (Browser.isSafariWebRTC() || Flashphoner.getMediaProviders()[0] === "MSE") {
    Flashphoner.playFirstVideo(remoteVideo, false, PRELOADER_URL).then(function () {
        playStream();
    });
    return;
}
```

The `playFirstVideo` function returns a Promise, which is resolved if a test video fragment was successfully played in video element by `PRELOADER_URL`.

3. If Websocket connection is establishing on page load, then playback starts by `SESSION_STATUS.ESTABLISHED` event (autoplay), it is necessary to use `playFirstVideo` and disable automatic unmute when stream is created

[code](#)

```

var options = {
    name: streamName,
    display: remoteVideo,
    flashShowFullScreenButton: true
};
...
if (autoplay) {
    options.unmutePlayOnStart = false;
}
stream = session.createStream(options).on(STREAM_STATUS.PENDING, function (stream) {
    ...
});
stream.play();

```

Stream publishing and playback in Low Power Mode

In Low Power Mode on iOS devices, user action is mandatory to publish or play media stream. Autoplay does not work in this mode. Use the function `playFirstVideo` to detect Low Power Mode: a Promise returned by the function will be rejected:

[code](#)

```

        if (Browser.isSafariWebRTC()) {
            Flashphoner.playFirstVideo(pDisplay, false, PRELOADER_URL).then(function() {
                playStream(participant, pDisplay);
            }).catch(function (error) {
                // Low Power Mode detected, user action is needed to start playback in this mode
                console.log("Can't automatically play participant" + participant.name() + " stream, use
Play button");
                onParticipantStopped(participant);
            });
        }

```