

# Publishing a stream using custom HTML5 video element

By default, WebSDK creates HTML5 video element automatically to display a stream captured. However, in some cases it is necessary to display a stream on custom video element. This can be done as follows

## 1. Add video element to example HTML page

```
...
    <div class="col-sm-6">
      <div class="text-center text-muted">Local</div>
      <div class="fp-Video">
        <div id="localVideo" class="display"><video id="customVideo" controls="controls"></video><
      /div>
    </div>
  ...
```

## 2. Create a container to set srcObject on stream creation

```
var mockLocalDisplay = $('<div></div>');
var mockLocalVideo = $('<video></video>', {id: 'mock-LOCAL_CACHED_VIDEO'});
mockLocalDisplay.append(mockLocalVideo);
```

## 3. Copy srcObject from the container to custom video element

```
var video = document.getElementById("customVideo");
stream = session.createStream({
  name: $('#streamName').val(),
  display: mockLocalDisplay.get(0)
}).on(STREAM_STATUS.PUBLISHING, function (stream) {
  var srcObject = mockLocalVideo.get(0).srcObject;
  video.srcObject = srcObject;
  mockLocalVideo.get(0).pause();
  mockLocalVideo.get(0).srcObject = null;
});
stream.publish();
```