

Azure Media Player - Quality of Experience (QoE) detector

Hive Video Exp is video streaming company that provides a cutting edge platform with high standards about User Experience. To achieve these standards, the Company needs a Monitor System that can detect problems as soon as they appear.

This task consists in implementing a small player plugin that collects information and telemetries from the Azure Media Player video player (AMP: <http://amp.azure.net/libs/amp/latest/docs/index.html>), sends them to a server for detecting if the viewer has a bad Quality of Experience of the stream.

The player is well documented and support multiple streaming protocols, depending on different browsers. In Microsoft Edge (Win), Firefox and Chrome (Win + OS X), it can be configured to use the DASH protocol, for which it is possible to access to a large amount of statistics and telemetries directly from the AMP developer API. Please use only this protocol during the test.

We prepared a test page configured to start a AMP player with a short demo video using the DASH protocol, and the goal is to implement the plugin stub to perform telemetries collection. The test page is accessible and downloadable [here](#). In the page, you'll find an AMP's plugin stub for you to fill in.

The first goal is to implement a plugin that generates an object containing the telemetries we are interested in. You can choose the best format for your telemetry object.

The info that needs to be collected is:

- video frame size
- available video bitrates
- bitrate switches
- number of buffering events
- time spent in buffering state

The following is some detailed information on how the code is organized:

This is the plugin stub:

```
(function () {
    amp.plugin('telemetry', function (options) {

        var player = this

        var init = function () {
            console.log("plugin telemetry initialized with player ",
player)
        }

        // initialize the plugin
        init();

    });
}).call(this);
```

The script that initializes the player with our plugin and loads the source is as follows:

```

var myPlayer = amp('vid1', {
    /* Options */
    "nativeControlsForTouch": false,
    autoplay: true,
    controls: true,
    width: "640",
    height: "400",
    techOrder: ['AzureHtml5JS'],
    plugins: {
        /* load our telemetry plugin */
        telemetry: {
            /* Options */
        }
    }
}, function () {
    console.log('Good to go!');
});

myPlayer.src([
    {
        src: "http://amssamples.streaming.mediaservices.windows.net/91492735-c523-432b-ba01-faba6c2206a2/AzureMediaServicesPromo.ism/manifest(format=mpd-time-csf)",
        type: "application/dash+xml"
    }
]);

```

The second part of the assignment involve implementing an HTTP service that receives the data calculates indexes regarding the playback QoE.

The indexes are:

- HIGHEST_BITRATE_POSSIBLE: It warns out if the bitrate chosen by the player is meant for a smaller player frame size
- TOO_MANY_BITRATE_SWITCHES: It warns out if the number of bitrate switches is higher than 2 every 10 secs.
- TOO_MANY_BUFFERING: It warns out if the number of buffering events longer than 500ms is higher than 3 per 30 secs or if there is any buffering event longer than 1s

Tips:

To reproduce bad QoE, we suggest to use chrome network throttling feature.

For any question please write to claudio.dicosmo@hivestreaming.com, kevin.easy@hivestreaming.com