P2SP SDK Integration Guide (Beta)

HTML integration by script tag

1. Download P2SP driver script.

```html
<script src="https://cdn.jsdelivr.net/npm/@mlytics/p2sp-sdk@0.3.0-b2/bundle/driver.min.js"></script>
```

2. Currently, our project only supports player video.js with HLS protocol. Please download core video.js library and script for HLS integration.

```html
<script src="https://cdn.bootcdn.net/ajax/libs/video.js/7.18.0/alt/video.core.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@mlytics/p2sp-sdk@0.3.0-b2/bundle/peripheral/videojs-hls.min.js"></script>
```

3. When page is loading, call driver.initialize() first.

```js
const driver = mlyp2sp.driver.initialize();
```

4. Replace HLS loader in video.js html5 settings so that any video download would use P2SP instead of original HTTP call.

```js
const player = window.videojs('video-tag-id', {
    ...
    html5: {
        hlsConfig: {
            liveSyncDurationCount: 7,
            loader: driver.integrations.HLSLoader
        }
    }
});
```

Here is full example.

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
       content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>P2SP test</title>
  <link href="https://cdn.bootcdn.net/ajax/libs/video.js/7.18.0/video-js.css" rel="stylesheet"/>
  <link href="https://unpkg.com/@videojs/themes@1/dist/fantasy/index.css" rel="stylesheet">
  <script
src="https://cdn.bootcdn.net/ajax/libs/video.js/7.18.0/alt/video.core.min.js">
</script>
```

```html
  <script src="https://cdn.jsdelivr.net/npm/@mlytics/p2sp-sdk@0.3.0-
b2/bundle/peripheral/videojs-hls.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/@mlytics/p2sp-sdk@0.3.0-
b2/bundle/driver.min.js"></script>
</head>
<body>
<div style="width: 800px">
  <video id="video" class="video-js vjs-theme-fantasy"></video>
</div>
<script>
    const driver = mlyp2sp.driver.initialize();

    window.videojs('video', {
        autoplay: true,
        controls: true,
        preload: 'auto',
        aspectRatio: '16:9',
        html5: {
            hlsConfig: {
                liveSyncDurationCount: 7,
                loader: driver.integrations.HLSLoader
            }
        },
        sources: [{
            src: 'https://wowza.peer5.com/live/smil:bbb_abr.smil/playlist.m3u8',
            type: 'application/vnd.apple.mpegurl'
        }]
    });
</script>
</body>
</html>
```

HTML integration with front-end framework (Vue.js)

1. Install P2SP driver.

```
npm i @mlytics/p2sp-sdk@0.3.0-b2
```

2. Install video.js library.

```
npm i video.js@7.18.0
```

Install video.js themes if needed.

```
npm i @videojs/themes@1.0.1
```

3. Call driver.initialize() first and bind video.js with our HLS loader plugin.
   Remember to do this just once.

```
import {driver} from '@mlytics/p2sp-sdk/driver';

driver.initialize();
```

```
import videojs from 'video.js';
import {VideojsHlsPlugin} from '@mlytics/p2sp-
sdk/driver/peripheral/player/videojs/hls';

VideojsHlsPlugin.register(videojs);
```

4. Replace HLS loader in video.js html5 settings so that any video download would
   use P2SP instead of original HTTP call.

```
import {HLSLoader} from '@mlytics/p2sp-sdk/driver/integration/streaming-hls';

const player = videojs('video-tag-id', {
    ...
    html5: {
        hlsConfig: {
            liveSyncDurationCount: 7,
            loader: HLSLoader
        }
    }
});
```

Here is full example.

```
<template lang="pug">
  video(:id="video.id" class="video-js vjs-theme-fantasy")
</template>

<script>
import 'video.js/dist/video-js.css';
import '@videojs/themes/dist/fantasy/index.css';

import videojs from 'video.js';
import {driver} from '@mlytics/p2sp-sdk/driver';
import {HLSLoader} from '@mlytics/p2sp-sdk/driver/integration/streaming-hls';
import {VideojsHlsPlugin} from '@mlytics/p2sp-
sdk/driver/peripheral/player/videojs/hls';

driver.initialize();
VideojsHlsPlugin.register(videojs);

export default {
    name: 'p2sp-test',
    data: () => ({
        video: {
            id: 'video',
            player: null
        }
    }),
    async mounted() {
        this.video.player = videojs(this.video.id, {
            liveui: true,
            autoplay: true,
```

```
            controls: true,
            preload: 'auto',
            aspectRatio: '16:9',
            html5: {
                hlsConfig: {
                    liveSyncDurationCount: 7,
                    loader: HLSLoader
                }
            },
            sources: [{
                src: 'https://wowza.peer5.com/live/smil:bbb_abr.smil/playlist.m3u8',
                type: 'application/vnd.apple.mpegurl'
            }]
        });
    },
    async beforeDestroy() {
        this.video.player.dispose();
    }
};
</script>
```