DASH Event Metadata API

DASH-IF Editor's Draft 22 May 2019

This version:

Editors:

Giridhar Mandyam (Qualcomm Technologies Inc.)

Paul Higgs (Huawei Inc.)

.

Abstract

This document specifies an API that a user agent or DASH client can expose for application access to DASH events. This builds upon Media Source Extensions.

Table of Contents

1. Introduction

The API's defined in this document are partial interfaces with respect to the `DASHEVent` event target. It is meant to cover both in-band ('emsg') and MPD-carriage of events.

2. DASHEvent Interface

WebIDL

```
[Constructor(SourceBuffer source)]
interface DASHEvent : EventTarget {
    readonly        attribute EventData             eventData;
                    attribute EventHandler          ondashevent;
    Promise         setEvents(Eventlist eventList);
};
```

2.1 Attributes

`eventData` of type `EventData`, readonly

When an event is encountered, the DASH client *MUST* extract the event data, and *MUST* initialize the object's `eventData` attribute to a string representation of the event data.

***`ondashevent`* of type EventHandler**

This event handler is invoked when a new DASH event arrives.

`setEvents()` of type Promise

This promise must include an `eventList` argument that enumerates all events in which the application is interested.

3. EventData Interface

WebIDL

```
interface EventData {
    readonly        attribute DOMString schemeIdURI;
    readonly        attribute DOMString value;
    readonly        attribute int? timescale;
    readonly        attribute DOMTimeStamp? presentationTime;
    readonly        attribute int? duration;
    readonly        attribute int? id;
    readonly        attribute ByteString messageData;
};
```

3.1 Attributes

`schemeIdURI` of type DOMString, readonly

The `schemeIDURI` attribute *MUST* return a URI that identifies the DASH event scheme.

`value` of type DOMString, readonly

The `value` attribute *MUST* return the value for the event stream element. The value semantics are defined by the owners of the scheme identified in the `schemeIdUri` attribute.

`timescale` of type int, readonly

The `timescale` attribute *MUST* return the timescale in units per second to be used for the derivation of different real-time duration values in the `EventData` elements. If not present, it shall be set to 1.

`presentationTime` of type DOMTimeStamp, readonly

The `presentationTime` attribute *MUST* return a value corresponding to the exact moment in the media presentation timeline that the event becomes active. If this attribute is not present then its value shall be set to NULL and the event is assumed to be active immediately.

`duration` of type int, readonly

The `duration` attribute *MUST* return the time for which the event is in effect starting from presentationTime. The value of the duration in seconds is the division of the value of this attribute and the value of the `timescale` attribute. If this attribute is not present then its value must be set to NULL and the event *MUST* be persisted until another DASH event is received.

`id` of type int, readonly

The `id` attribute *MUST* return an identifying value for this event. If this value is not present then its value must be set to NULL.

`messageData` of type DOMString, readonly

The `messageData` attribute *MUST* return the event message data payload.

4. EventList Interface

WebIDL

```
dictionary EventList {
    DOMString[]     desiredSchemeIdURI;
};
```

`EventList` contains one or more valid event scheme URI's.

4.1 Members

`desiredSchemeIdURI` of type DOMString[]

`desiredSchemeIdURI` is an array of valid DASH event scheme URI's.

5. Example

EXAMPLE 1

```
<html>
<body>

<script>
  function onSourceOpen(videoTag, e) {
    var mediaSource = e.target;
    if (mediaSource.sourceBuffers.length > 0)
        return;
    try {
        dashevent = new DashEvent(mediaSource);
        dashevent.setEvents(["schemeURI1","schemeURI2"]).then(
            {
            console.log('Desired event list set');
            }
        }
        catche (e)
            {
            console.error('Failed to create Dash event handler due to:  ' + e);
            return;
            }
    dashevent.ondashevent = dashEventHandler;
    function dashEventHandler(event){
        }
    }
</script>
<video id="v" autoplay> </video>
<script>
  var video = document.getElementById('v');
  var mediaSource = new MediaSource();
  mediaSource.addEventListener('sourceopen', onSourceOpen.bind(this, video));
  video.src = window.URL.createObjectURL(mediaSource);
</script>
</body>
</html>
```

A. References

A.1 Informative references

[WEBIDL]

Web IDL. Boris Zbarsky. W3C. 15 December 2016. Editor's Draft. URL: https://heycam.github.io/webidl/