

Updating the document in sync with a media playing

Mixing JSON cue content and track and cue events makes the synchronization of elements in the HTML document (while the video is playing) much easier.

Example of track event listeners that use JSON cue contents

Here is a small code extract that shows how we can get the JSON content of a cue when the video reaches its start time. We do this in a `cuechange` listener attached to a `TextTrack`:

```
textTrack.onscuechange = function () {  
    // "this" is the textTrack that fired the event.  
    // Let's get the first active cue for this time segment  
    var cue = this.activeCues[0];  
    var obj = JSON.parse(cue.text);  
    // do something  
}
```

Here is [a very impressive demo by Sam Dutton](#) that uses JSON cues containing the latitude and longitude of the camera used for filming the video to synchronize two map views: every time the active cue changes, a Google map and a Google street view are updated.

XML map track demo



Example of a cue content from this demonstration:

```
{"lat":37.4219276, "lng":-122.088218,"t":1331363000}
```

Cue events and cue content:

We can get a handle to a cue DOM object using the techniques we have seen previously, or by using the new HTML5 `TextTrack.getCueById()` method.

```
var videoElement = document.querySelector("#myvideo");
var textTracks = videoElement.textTracks; // one for each track
element
var textTrack = textTracks[0]; // corresponds to the first
track element
// Get a cue with ID="wikipedia"
var cue = textTrack.getCueById("Wikipedia");
```

And once we get a handle to a cue, it is possible to add some event listeners to that cue:

```
cue.onenter = function(){
    // display something, play a sound, update any DOM
    element...
};
cue.onexit = function(){
    // do something else
};
```

If the `getCueById` method is not implemented (this is the case in many browsers, as at November 2015), it's easy to use a small polyfill:

```
// for browsers that do not implement the getCueById() method
// let's assume we're adding the getCueById function to a
TextTrack object
//named "track"
if (typeof track.getCueById !== "function") {
    track.getCueById = function(id) {
        var cues = track.cues;
        for (var i = 0; i !== track.cues.length; ++i) {
            if (cues[i].id === id) {
                return cues[i];
            }
        }
    }
}
```

```
13.     };  
14.     }
```

EXAMPLE THAT DISPLAYS A WIKIPEDIA PAGE AND A GOOGLE MAP
WHILE A VIDEO IS PLAYING

[Try the example at JSBin](#)

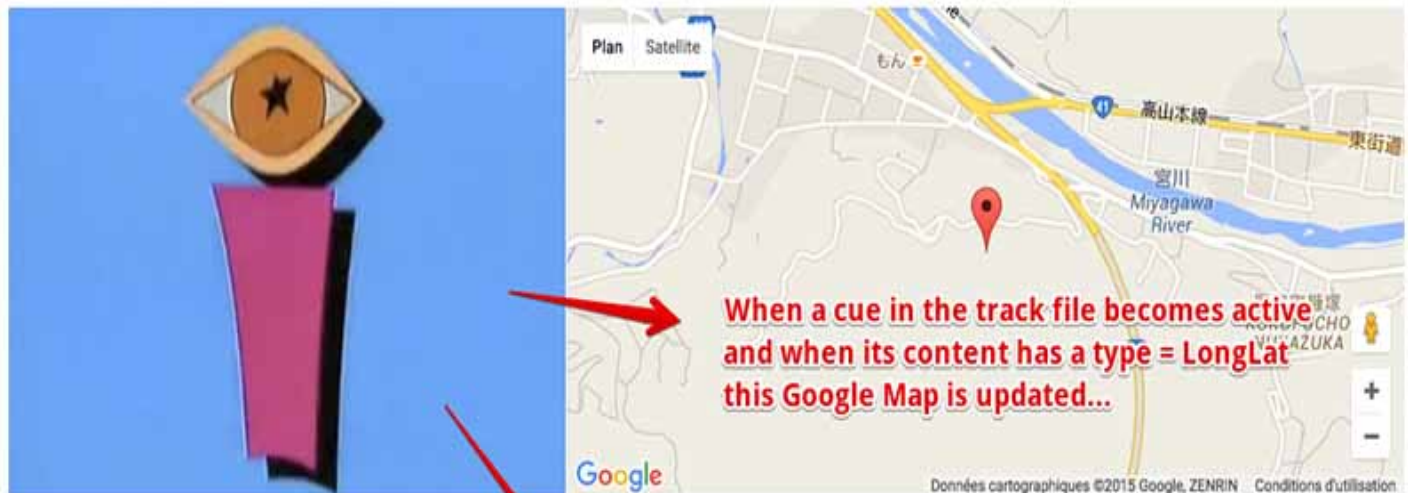
Example of a video in sync with an iframe and a Google Map

We use for this a [webVTT metadata file that contains JSON objects](#):

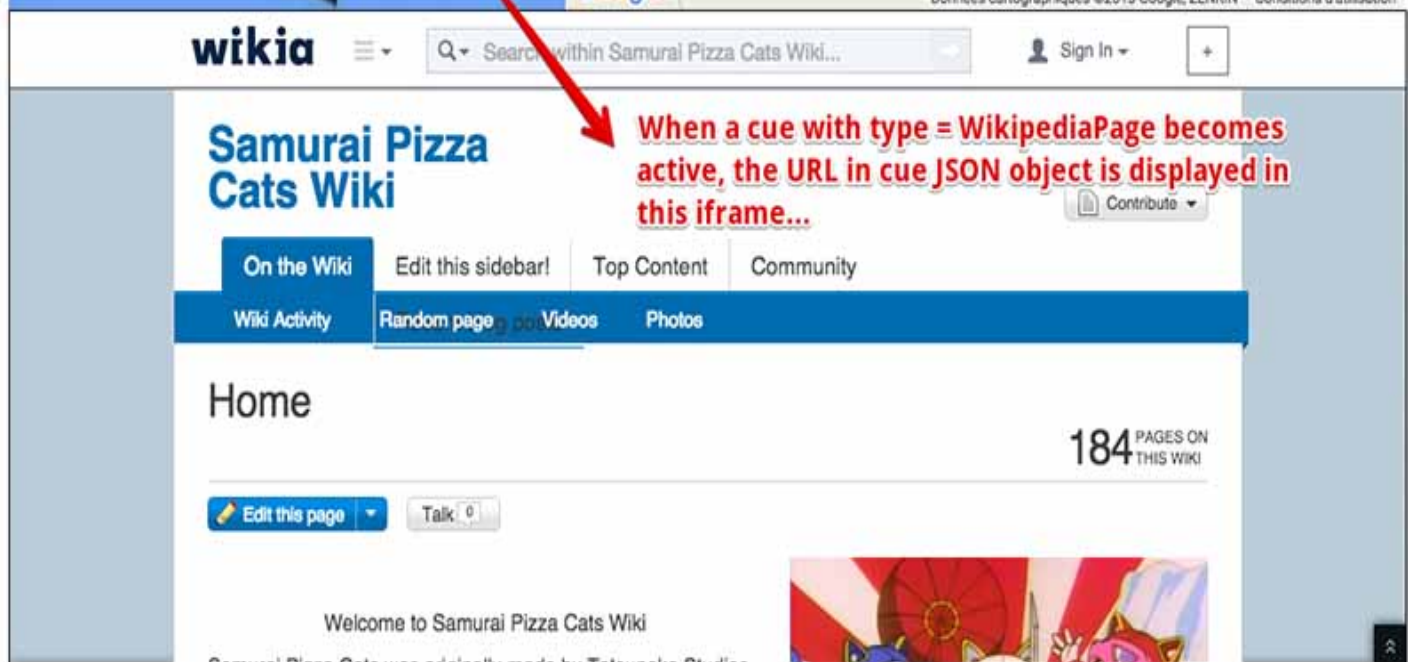
```
2
00:07.810 --> 00:09.221
{
  "type": "WikipediaPage",
  "url" : "http://samuraipizzacats.wikia.com/wiki/Samurai_Pizza_Cats_Wiki"
}

3
00:11.441 --> 00:14.441
{
  "type": "LongLat",
  "lat" : "36.198269",
  "long": "137.2315355"
}
```

The video uses a <track> file whose content looks like this one.



When a cue in the track file becomes active and when its content has a type = LongLat this Google Map is updated...



When a cue with type = WikipediaPage becomes active, the URL in cue JSON object is displayed in this iframe...

Wikipedia URL: http://samuraipizzacats.wikia.com/wiki/Samurai_Pizza_Cats_Wiki

HTML code extract:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example syncing element of the document with video
  metadata in webVTT file</title>
</head>
<body >
  <main>
    <video id="myVideo" controls<crossorigin="anonymous" >
10.   <source src="http://mainline.i3s.unice.fr/mooc/samuraiPizzacat.mp4"
11.         type="video/mp4">
12.   </source>
    <track label="urls track"
          src="http://...../SamuraiPizzaCat-metadata.vtt"
          kind="metadata" >
    </track>
    </video>
    <div id="map"></div>
  </main>
  <aside>
    <iframe sandbox="allow-same-
origin" id="myIframe" > </iframe>
  </aside>
24. <h3>Wikipedia URL: <span id="currentURL"> Non défini </span>
    </h3>
    <script src="http://maps.google.com/maps/api/js?sensor=false">
    </script>
    ...
```

JavaScript code:

```
window.onload = function() {
```

```
var videoElement =document.querySelector("#myVideo");
var myIFrame =document.querySelector("#myIframe");
var currentURLSpan
= document.querySelector("#currentURL");
var textTracks =videoElement.textTracks; // one for each
track element
var textTrack = textTracks[0]; // corresponds to the first
track element
```

```
10. // change mode so we can use the track
textTrack.mode = "hidden";
// Default position on the google map
var centerpos = newgoogle.maps.LatLng(48.579400,7.7519);
```

```
// default options for the google map
var optionsGmaps = {
    center:centerpos,
    navigationControlOptions: {style:
        google.maps.NavigationControlStyle.SMALL},
19. mapTypeId:google.maps.MapTypeId.ROADMAP,
    zoom: 15
};
```

```
// Init map object

var map = newgoogle.maps.Map(document.getElementById("map"),
    optionsGmaps);
```

```
// cue change listener, this is where the synchronization
between
// the HTML document and the video is done
textTrack.onscuechange = function (){
    // we assume that we have no overlapping cues
    var cue = this.activeCues[0];
    if(cue === undefined) return;
    // get cue content as a JavaScript object
    var cueContentJSON =JSON.parse(cue.text);
    // do different things depending on the type of sync
(wikipedia, gmap)
    switch(cueContentJSON.type) {
```

```

        case 'WikipediaPage':
            var myURL = cueContentJSON.url;
            var myLink = "<a
42. href=\"\" + myURL + \">\" + myURL + "</a>";
            currentURLSpan.innerHTML = myLink;
            myIFrame.src = myURL; // assign url to src
            property
            break;
            case 'LongLat':

drawPosition(cueContentJSON.long, cueContentJSON.lat);
            break;
        }
50. };

function drawPosition(long, lat) {
    // Make new object LatLng for Google Maps
    var latlng = new google.maps.LatLng(lat, long);

    // Add a marker at position
    var marker = new google.maps.Marker({
58. position: latlng,
        map: map,
        title: "You are here"
    });
    // center map on longitude and latitude
    map.panTo(latlng);
}
};

```

All the work is done on a `cuechange` event listener, *lines 27-47*. As we have only one track, we set its mode to "hidden" (*line 10*) in order to be sure that it will be loaded and that playing the video will fire `cuechange` events on it. The rest is just Google map code and classic DOM manipulation for updating some HTML content (a span that will display the current URL, *line 40*).