RadioKit JS Toolkit Documentation

Release 1.0

RadioKit Ltd

May 24, 2017

Contents

RadioKit JavaScript Toolkit is intended to be a high-level set of libraries that can be used to assemble web-based applications based on the RadioKit infrastructure.

So far it contains one library, for doing playback of RadioKit-based channels.

CHAPTER 1

Contents

Playback

Introduction

Playback Toolkit can be used to easily play channels broadcasted from RadioKit systems through the website.

In the long run it is going to encapsulate a lot of sophisticated features that are hard to implement manually, such as codecs and protocol negotiation, handling workarounds for different browsers, sending back statistics etc.

This is the recommended way to play back channels (streams) in the web browser.

Demo

Demo is available at http://radiokit.github.io/toolkit-js-playback-demo

Compatibility

So far it is known to work with recent versions of

- Firefox,
- Chrome,
- Opera,
- Safari (on Mac),
- Edge.

Mobile support is in progress.

At the moment there's no automatic detection of browsers in the code. It is your responsibility to warn user if one uses unsupported browser.

Prerequisites

Playback Toolkit has some prerequsities.

At the moment there's no automatic detection of these features in the code. It is your responsibility to warn user if one uses unsupported browser.

Promises

Playback Toolkit assumes that Promises are supported by the browser but does not bundle any polyfill for the older browsers by itself.

If you want to ensure that it works properly you have to include such polyfill on your own in the app.

HTML5 Audio

Playback Toolkit works only on browsers that support HTML5 Audio tag and are capable to decode MP3 or Ogg/Opus format. Virtually all browsers in use nowadays should handle this properly.

There is no flash fallback and it is not planned at the moment as browsers that may require it are so old that you should instruct your users to upgrade anyway.

HTML5 doctype

It is known that some browsers have issues with HTML media players if the document is not valid HTML5. Please ensure that you output valid HTML5 doctype and conform to other requirements of the standard.

Channel IDs and Access Tokens

Before you use the Playback Toolkit you must obtain channel IDs (strings in the UUIDv4 format that uniquely identify radio channels) and Access Token (string that authorizes your application) from RadioKit administrators.

Sample usage

The sample below shows how to use player at the website.

At the moment it covers the whole public API available so there's no separate reference:

```
<!DOCTYPE html>
<html>
<head>
  <!-- Required by HTML5 -->
  <title>RadioKit JavaScript Toolkit Playback Demo</title>
  <meta charset="utf-8">

  <!-- Include Promise polyfill for older browsers -->
  <script src="core-promise.js"></script>

  <!-- Include browser version of the toolkit -->
  <script src="radiokit-toolkit-playback-1.0.2.js"></script>

  <script>
  function init() {
```

```
    // Replace this with your channel ID
    var channelId = "fd9a7d1c-a387-40a0-b876-2799668d6f9d";

    // Replace this with your access token
    var accessToken = "demo";

    // Initialize global player instance
    window.player = new RadioKitToolkitPlayback.Channel.Player(channelId,␣
↪accessToken);

    // React to new track
    window.player.on('track-playback-started', function(track) {
      document.getElementById('track-id').innerHTML = 'TRACK ID: ' + track.getId();

      // Asynchronously extract track metadata
      track.getInfoAsync()
        .then(function(trackInfo) {

          // Name
          document.getElementById('track-name').innerHTML = 'NAME: ' + trackInfo.
↪getName();

          // Metadata
          var metadataInfo = trackInfo.getMetadata();
          var metadataText = 'METADATA:<ul>';
          metadataKeys = Object.keys(metadataInfo);
          for(var i = 0; i < metadataKeys.length; i++) {
            metadataText += '<li><b>' + metadataKeys[i] + '</b>: ' +␣
↪metadataInfo[metadataKeys[i]] + '</li>';
          }
          metadataText += '</ul>';

          document.getElementById('track-metadata').innerHTML = metadataText;

          // Affiliates
          var affiliatesInfo = trackInfo.getAffiliates();
          var affiliatesText = 'AFFILIATES:<ul>';
          affiliatesKeys = Object.keys(affiliatesInfo);
          for(var i = 0; i < affiliatesKeys.length; i++) {
            affiliatesText += '<li><b>' + affiliatesKeys[i] + '</b>: ' + JSON.
↪stringify(affiliatesInfo[affiliatesKeys[i]]) + '</li>';
          }
          affiliatesText += '</ul>';

          document.getElementById('track-affiliates').innerHTML = affiliatesText;
        })
        .catch(function(reason) {
          throw reason;
        });
    });


    // React to track position changes
    window.player.on('track-position', function(track, position, duration) {
      document.getElementById('track-position').innerHTML = 'POSITION: ' + position +
↪' / ' + duration + ' ms';
    });
  }
```

```
  // Handler for play/stop button
  function toggle() {
    var button = document.getElementById('button');

    if(window.player.isStarted()) {
      button.innerHTML = "PLAY";
      window.player.stop();
    } else {
      button.innerHTML = "STOP";
      window.player.start();
    }
  }


  // Handler for volume buttons
  function setVolume(volume) {
    window.player.setVolume(volume);
  }
  </script>
</head>

<body onload="init()" style="text-align: center">
  <button onclick="toggle()" id="button" style="width: 400px; height: 200px; font-
↪size: 24pt">PLAY</button>

  <br><br>

  <button onclick="setVolume(0.25)" style="width: 97px; height: 50px; font-size: 11pt
↪">VOL 25%</button>
  <button onclick="setVolume(0.50)" style="width: 97px; height: 50px; font-size: 11pt
↪">VOL 50%</button>
  <button onclick="setVolume(0.75)" style="width: 97px; height: 50px; font-size: 11pt
↪">VOL 75%</button>
  <button onclick="setVolume(1.0)" style="width: 97px; height: 50px; font-size: 11pt">
↪VOL 100%</button>

  <br><br>

  <div id="track-id" style="width: 400px; text-align: center; margin: 0 auto"></div>
  <br>

  <div id="track-position" style="width: 400px; text-align: center; margin: 0 auto"></
↪div>
  <br>

  <div id="track-name" style="width: 400px; text-align: center; margin: 0 auto"></div>
  <br>

  <div id="track-metadata" style="width: 400px; text-align: left; margin: 0 auto"></
↪div>
  <br>

  <div id="track-affiliates" style="width: 400px; text-align: left; margin: 0 auto"></
↪div>
</body>
</html>
```

Installation

NPM

If you want to use Playback Toolkit with your NPM-based projects, type

```
npm install --save radiokit-toolkit-playback
```

Then you can use it in your code like this:

```
import { Channel } from 'radiokit-toolkit-playback';

let channelId = "fd9a7d1c-a387-40a0-b876-2799668d6f9d";
let accessToken = "demo";
let player = new Channel.Player(channelId, accessToken);
```

Browser

You can download the most recent browser bundle from the dist/browser subdirectory of the repository http://bitbucket.org/radiokit/toolkit-js-playback

Then you can just embed it in the HTML using regular script tag:

```
<script src="radiokit-toolkit-playback-1.0.2.js"></script>
```

once it's loaded, the global window.RadioKitToolkitPlayback object is available.

Support

If you need any support while developing, please do not hestitate to contact us at info@radiokit.org.