글로벌 온라인 미디어 스트리밍 Best Practice

Part 4. Building a Streaming HTML5 Player

글.
정영석 라임라이트 네트워크스
Advanced Service Architect

글로벌 미디어 스트리밍 서비스 시리즈

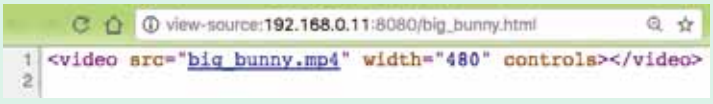
1. Global Online Media Streaming
2. HTTP Streaming, Current & Next
3. Codec, Current & Next
4. Building a Streaming HTML5 Player

이번 회차는 글로벌 온라인 미디어 스트리밍 시리즈의 마지막으로 Streaming Player에 대해 알아보고자 한다. Video Player는 비디오 시청자와 직접 소통하게 되는 인터페이스일 뿐만 아니라 클라이언트 환경을 고려한 영상 품질 선택, 콘텐츠 보호, 광고 처리, 사용자 분석 및 서버 사이드의 에러 처리 등 다양한 역할을 담당하게 된다. 이와 같은 기능을 처리하기 위해 네이티브 앱이나 플러그인 기반 앱을 많이 활용해 왔으나, HTML5를 지원하는 브라우저가 다양해 지면서 웹 페이지 내에 HTML5를 활용하여 Video Player를 개발하는 빈도가 높아지고 있다. HTML5를 이용해 플레이어를 개발하는 방법의 특징을 살펴보고 주의 해야 될 사항은 어떤 것들이 있는지 알아보자.

HTML5 Video Element를 이용한 Video Player

HTML5 Video란?

HTML5 Video는 왼편의 그림과 같이 웹 페이지 상에 비디오를 보여주기 위한 HTML5 Element 이다. Video Element를 이용해 매우 간단하게 Video Player를 사용자에게 제공해 줄 수 있다.

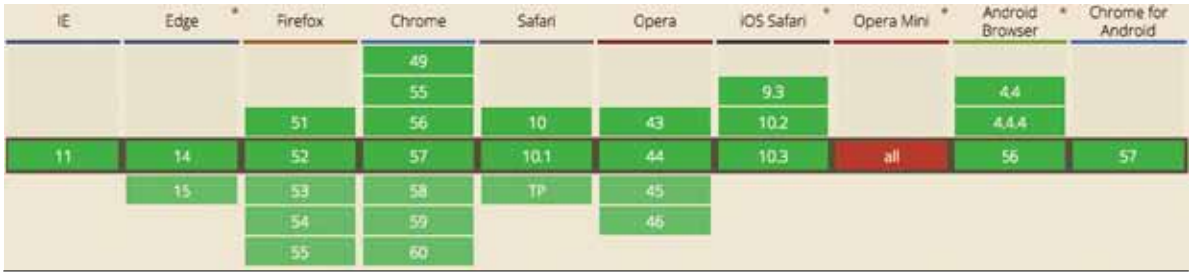


이와 같이 Video Player가 보이는 웹 페이지를 만들기 위한 HTML 코드를 살펴보자.

Video 태그 안에 재생할 영상 파일, 영상 화면의 크기 및 버튼을 표시하기 위한 옵션만 표시해도 그럴듯한 플레이어를 웹 페이지 상에 표시해 줄 수 있다. 이렇듯 콘텐츠 제작자는 HTML5 Video Element를 이용하면 사용자에게 쉽게 Video Player를 제공해 줄 수 있다. Video Player에 대한 실제 구현은 브라우저 벤더에서 HTML5 스펙에 따라 구현해 놓았기 때문에 우리는 Video Element를 통해 입맛에 맞도록 변경해가며 사용할 수 있다. 물론 브라우저 벤더별로 HTML5 Video를 지원하기 시작한 시점과 구현 방법이 다르기 때문에 브라우저별 확인이 필요하다.

HTML5 Video를 지원하는 브라우저

http://caniuse.com에서는 HTML5 Element별 브라우저 지원 여부를 한눈에 확인할 수 있도록 정보를 제공해준다. 아래 그림은 HTML5 Video를 지원하는 브라우저 버전을 보여준다. 다행히 HTML5 Video는 일부 모바일 브라우저를 제외한 대부분의 주요 브라우저에서 지원된다.



HTML5 Video Element를 지원하는 브라우저

브라우저에서 HTML5 Video를 지원한다고 해서 무조건 영상이 재생되는 것은 아니다. 브라우저가 해석할 수 있는 영상 포맷을 사용하는지 확인이 필요하다. HTML5에서는 MP4, WebM, Ogg 포맷을 지원하며, 브라우저별 지원되는 포맷이 다르므로 서비스하고자 하는 영상 포맷에 대한 브라우저 지원 여부를 확인할 필요가 있다.

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (from Opera 25)	YES	YES

영상 포맷별 브라우저 지원 여부 / 출처 : www.w3schools.com

MP4는 H.264 코덱과 함께 사용되며 대부분의 브라우저에서 지원되는 장점이 있으나 유료 비디오 서비스를 할 경우 라이센스 비용을 고려해야 하므로 MP4 사용 시 라이센스 정책에 대한 정확한 이해가 필요하다. 반면 WebM과 Ogg의 경우 라이센스 고민 없이 사용할 수 있는 포맷이지만 지원되는 브라우저가 제한적이므로 서비스 모델과 대상 브라우저에 따라 포맷을 고려할 필요가 있다.

HTML5 Video 기능

HTML5 Video에서는 기본적으로 영상 재생뿐만 아니라 자막, 비디오 포스터, 대체 영상 파일 지정 등 다양한 기능을 제공한다. 샘플과 함께 일부 기능들을 살펴보자.

video 태그 안의 poster 속성은 영화 포스터를 보여주듯이 비디오 시작 전에 보여줄 이미지를 설정할 수 있다. source 태그는 다수의 영상 파일을 명시해 주기 위해 사용된 태그이다. 여기서는 브라우저가 webm 타입의 비디오를 재생할 수 없을 경우 mp4 타

정영석
라임라이트 네트웍스
Advanced Service
Architect

입의 비디오를 재생하도록 두 개의 source 태그가 사용되었으며 혹시 webm과 mp4 모두 재생이 불가능할 경우 "Your browser does not support the video tag"라는 텍스트를 웹 페이지에 표시되도록 처리하였다. 각각의 source 태그는 src 속성으로 비디오 파일의 위치를 표시한다. type 속성에는 "video/webm"과 "video/mp4"를 표시해 놓았는데 이 부분은 생략해도 비디오 재생은 가능하다. 하지만 type 속성을 명시하지 않으면 브라우저에서 직접 비디오 콘텐츠 일부를 다운받아 지원 가능한 포맷인지 확인하는 절차를 거치게 되므로 경우에 따라서 비디오의 첫 프레임이 화면에 나오는 시점이 지연될 수 있다. 따라서 비디오 포맷을 알고 있을 경우 type 속성을 명시해주는 것이 바람직하다.

옆의 화면은 자막 및 재생 관련 기능을 활용한 예이다. UI상으로 자막이 추가되고 자막을 끄거나 다른 언어로 변경할 수 있는 메뉴가 생겼다. 이와 같은 자막 기능은 track 태그를 이용해 처리가 가능하다. 우선 src 속성을 통해 자막 파일을 정의하고 kind와 srclang을 통해 영어 자막이 있다는 것을 브라우저에 알려 준다. 영어 자막을 default 속성으로 처리하여 기본적으로는 영어 자막이 보이도록 하였고, 한글 자막 track도 준비하여 kr 선택 시 한글이 보이도록 하였다.

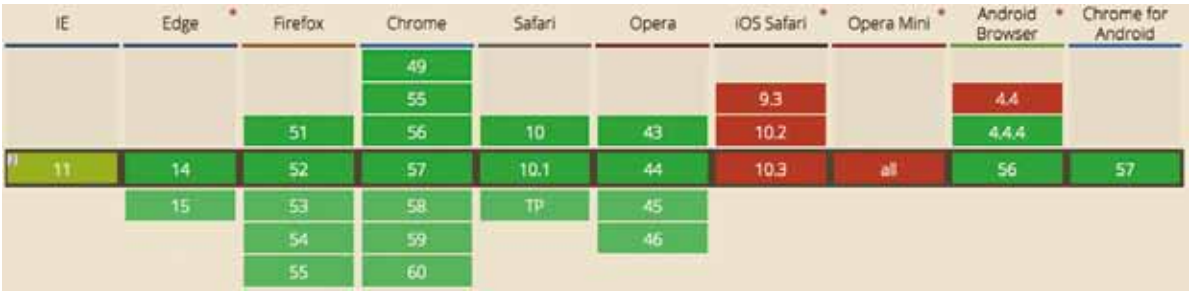
video 태그에 loop과 autoplay가 추가된 것도 보인다. loop은 영상 재생을 마치면 처음으로 돌아가 반복 재생하도록 하는 기능이다. autoplay는 웹 페이지 로드 시 비디오를 자동으로 재생하도록 하는 기능이다. autoplay에 대응되는 기능으로 preload라는 속성이 있는데 이는 비디오를 자동 재생시키지 않고 영상 데이터만 미리 받아오도록 처리하거나 Play 버튼을 누르기 전까지는 영상 데이터를 받지 않도록 처리하는데 사용된다.

Media Source Extension의 등장

지금까지 HTML5 Video를 이용해 Video Player를 만드는 방법을 살펴보았다. 스트리밍 영상도 HTML5 Video로 처리가 가능할까? 결론부터 말하면 HTML5 Video Element 자체로는 HLS나 MPEG-DASH와 같은 HTTP 스트리밍 영상은 재생할 수 없다. 즉, HTTP 기반 라이브 스트리밍이나 사용자 네트워크 환경을 고려한 Adaptive Bitrate Streaming을 제공해 줄 수 없다는 것이다. (ABR에 대한 자세한 설명은 방송과 기술 2017년 3월호 HTTP Streaming Current & Next를 참고하기를 바란다) 이 부분을 보완하기 위해 MSE(Media Source Extensions)라는 W3C 스펙이 나오게 되었다.

MSE는 Javascript가 Byte Stream을 웹 브라우저의 미디어 코덱에 전달할 수 있도록 해준다. 즉, Javascript에서 미디어 스트림을 만들어 HTML5 Video 태그에 전달할 수 있게 된 것이다. MSE의 등장과 함께 순수 HTML5와 자바스크립트만으로 MPEG-DASH

나 HLS와 같은 HTTP Streaming 비디오에 대한 재생이 가능하게 되었다. 다만 MSE를 지원하지 않는 브라우저에서는 정상적인 동작을 기대할 수 없다. 다행히도 일부 모바일 브라우저를 제외한 대부분의 최신 브라우저에서는 MSE를 지원하는 노선을 택하고 있다.



Media Source Extensions가 지원되는 브라우저 버전, 출처 : caniuse.com

HTML5 기반 Video Player의 동작 과정을 살펴보면 다음과 같은 순서로 영상 재생이 이루어진다.

① Manifest 파일을 내려받아 비디오 재생 정보를 분석한다. 이 Manifest 파일은 어떤 비트레이트의 영상을 사용할 수 있는지, 오디오/영상/자막이 저장된 위치 등을 포함한다.
② 클라이언트의 네트워크 환경을 고려하여 상황에 최적화된 영상 세그먼트를 내려받는다. (Javascript 담당)
③ 내려받은 영상 세그먼트를 MSE 버퍼로 넘긴다. (Javascript 담당)
④ HTML5 Video에서 MSE 버퍼에 있는 영상을 디코딩 후 화면에 보여준다.

넷플릭스 및 유튜브는 일찌감치 HTML5 기반 Player로 서비스를 제공해 왔으며, 최근에는 대부분의 Streaming Player 벤더에서도 HTML5 기반 Player를 우선적으로 제공하고 MSE가 지원되는 않는 브라우저를 사용하는 사용자에게는 Flash Player로 처리되도록 하는 추세이다.

MSE을 이용한 Player 개발

HTML5와 MSE를 이용해 Video Player를 만들기 위해서는 Javascript에 대한 이해도가 필요하다. 옆의 코드는 HTML5 Video와 MSE를 이용해 개발한 Sample HTML5 Video Player이다. 코드를 간단히 살펴보면 먼저 HTML5 video 태그를 만들어 놓은 후 미디어 정보가 저장된 MediaSource 인스턴스를 HTML5 video에 연결 시켜 준다. 이후 서버에서 영상 파일을 내려받아 MediaSource 버퍼에 저장한 후 HTML5 Video를 재생시킨다.

MSE를 통해 기능 구현상의 유연성이 증가한 반면 구현량을 봤을 때는 직접 처리해줘야 되는 부분이 늘게 되었다. 즉, MSE를 이용해 MPEG-DASH나 HLS를 재생시키고 Adaptive Bitrate Streaming을 구현하는 것은 만만치 않은 작업량을 필요로 하게 된다.

정영석
라임라이트 네트웍스
Advanced Service
Architect

고맙게도 DASH-IF 및 일부 그룹에서 MSE 기반 HTTP Adaptive Streaming Player를 오픈소스로 제공해 주고 있다. 널리 사용되고 있는 MSE 기반 오픈소스 플레이어에 대해 알아보도록 하자.

오픈소스 기반 HTML5 Player 검토

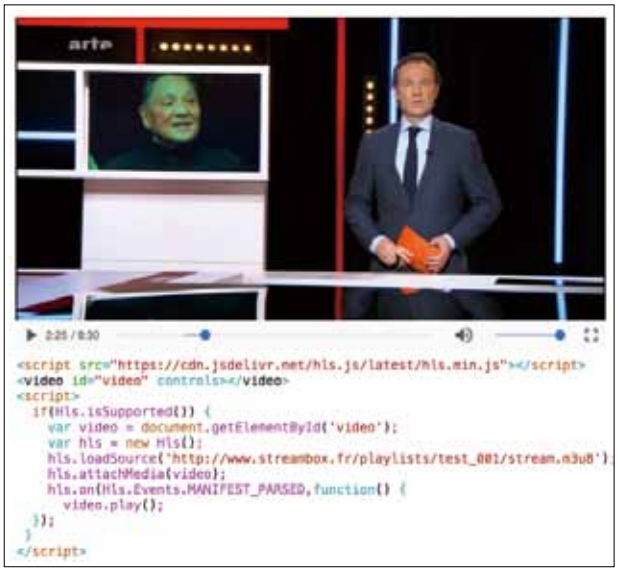
HTML5 Streaming Player를 만들 때 주로 거론되는 오픈소스는 Video.js, Dash.js, hls.js 등이 있다. 이 프로젝트들은 모두 Github에서 활발하게 업데이트되고 있으며 다양한 서비스에서 수년간 검증되어 왔다. 구현 방식, 지원 기능 및 주요 강점은 모두 다르지만 MSE 기반으로 만들어진 Streaming HTML5 Player에 사용되는 Javascript 라이브러리라는 점은 동일하다. 각 오픈소스의 별 특징에 대해 살펴보자

dash.js

DASH-IF(DASH 연동 스펙 및 사용 가이드 등을 제공해 주는 그룹)에서 주도하고 있는 오픈소스 프로젝트이며 MPEG-DASH에 대해 재생, Adaptive Streaming, DRM, 광고 삽입 등에 대한 처리를 제공해 준다. 타 오픈소스 프로젝트에 비해 일찍 시작한 편이라 프로덕션 환경에 널리 사용되고 있으며 다양한 테스트 케이스 등이 준비되어 있다. 또한 DASH-IF로부터 MPEG-DASH 재생에 대한 Best Practice를 제시해주고 있어 MPEG-DASH Player를 개발하고자 한다면 검토해봐야 할 프로젝트이다.

hls.js

DailyMotion에서 주도적으로 이끌어 오던 오픈소스 프로젝트로 HLS 재생, Adaptive Streaming 등의 기능을 제공한다. Video.js(videojs-hls.js)뿐만 아니라 다수의 오픈소스 및 상용 Player에 통합되어 사용되고 있는 프로젝트로 버전이 1.0에 도달 못했지만(현재 v0.75까지 릴리즈) 널리 사용되고 있는 MSE 기반 HTTP Streaming 라이브러리이다. 동작 원리는 MPEG-2 Transport Stream을 ISO BMFF(MP4) Fragment로 변경한 후 MSE 버퍼로 넘겨 재생이 되도록 하는 구조이다. (fmp4 포맷으로 패키징 된 HLS Segment도 지원한다) 최근 video-dev 라는 오픈소스 그룹으로 소속이 변경되면서 어떤 횡보를 걷게 될지 지켜볼 필요가 있다.



Video.js

Brightcove에서 주도적으로 후원하고 있는 오픈소스 프로젝트로 HLS 및 MPEG-DASH의 재생이 가능하고 Adaptive Streaming, 광고 삽입, 다양한 플레이어 스킨을 제공한다. 또한, 플러그인 프로젝트만 해도 40개 이상이 진행되고 있어 활용 가능한 기능이 많다는 점도 큰 장점이다.

옆의 샘플의 경우에는 MPEG-DASH 재생을 위해 video-dash.js 플러그인과 dash.js를 활용하였다. 이렇듯 video.js의 경우 다양한 기능을 Player에 연동시키기 용이한 구조로 되어 있다. HLS 및 MPEG-DASH를 동시에 지원해야 되고 Player 디자인이 중요하다면 Video.js 프로젝트를 확인해 보길 권장한다.

```
<video id=example-video width=600 class="video-js vjs-default-skin" controls></video>
<link href="https://unpkg.com/video.js/dist/video-js.css" rel="stylesheet">
<script src="video.js"></script>
<script src="dash.all.min.js"></script>
<script src="videojs-dash.js"></script>
<script>
var player = videojs('example-video');

player.ready(function() {
  player.src({
    src: 'http://dash.edgesuite.net/envivio/EnvivioDash3/manifest.mpd',
    type: 'application/dash+xml'
  });
  player.play();
});
</script>
```

오픈소스를 이용하여 Video Player를 개발하는 것은 개발 시간을 단축시킬 수 있다는 점에서 분명 매력적인 선택이다. 하지만 서비스 운영 중 문제 발생시 신속히 트러블슈팅하여 적시에 패치 버전을 내놓을 수 있는지, 새로운 요구사항이 생겼을 때 원활히 기능 추가가 가능할지에 대한 고민이 필요하다. 비디오 스트리밍 서비스를 할 경우 Video Player는 서비스의 중요한 부분을 담당하는 컴포넌트이다. 따라서 오픈소스를 이용하여 Video Player를 개발할 경우 조직 내에 일부 개발자가 오픈소스 커미터로 활동하는 것이 가장 바람직하다. 여건이 안될 경우 오픈소스 구조에 대해 이해할 수 있는 인력을 양성하여 문제를 진단하고 보완 방향에 대해 오픈소스 진영에 요청할 수 있는 수준까지 준비하는 것이 필요하다.

Video Player 개발을 위한 개발 인력 확보가 어렵거나 타임투마켓이 중요한 서비스의 경우 상용 플레이어에 대한 검토가 필요하다. Video Player 벤더들도 대부분 HTML5 기반의 Player를 제공하며 위에서 소개한 오픈소스를 기반으로 Video Player를 제공하는 곳도 많다. 상용 Video Player 선택 시 지원 가능한 스트리밍 프로토콜, DRM, 광고 삽입, 기술지원, 비용에 대해 알아보고 부수적으로 Analytics 수준에 대해서도 비교하기 바란다. Analytics는 콘텐츠 별 재생 관련 통계와 사용자 관점에서 버퍼링을 언제, 어떤 빈도로 겪었는지 등에 대해 파악할 수 있게 시각화 정보를 제공해 준다. 이런 QoE 정보는 서비스의 영상 정책 수립(Bitrate) 및 콘텐츠 전송 품질 판단에 객관적인 데이터로 활용이 가능하다.

마치며

지금까지 HTML5 Video와 MSE를 이용해 Video Player를 개발하는 방법에 대해 살펴보았다. HTML5가 보편화 됨에 따라 온라인 Video Player의 기반 기술에 많은 변화가 있었다. 네이티브나 플러그인 기반 Video Player의 경우 플랫폼별 Video Player를 만들어야 했지만, HTML5를 이용해 Video Player를 개발할 경우 한 벌의 코드로 다수의 디바이스에서 동작되는 Video Player를 만들 수 있는 세상이 되었다. 아직 모든 브라우저가 HTML5 Video와 MSE를 완벽히 지원하지는 않지만 W3C 및 다수의 표준화 그룹에서는 표준화된 Video 기술 확산을 위해 지속적으로 노력하고 있고 대부분의 브라우저 벤더들도 변화에 순응하고 있는 것으로 보인다. 이에 따라 HTML5 Video를 활용한 개발은 현시점에서 택할 수 있는 최고의 옵션으로 생각된다.

이상으로 글로벌 미디어 스트리밍 서비스의 연재를 마치도록 하겠습니다. 지금까지 읽어주신 독자 여러분께 감사를 드리며, 인터넷 스트리밍 관련 기술에 대해 익숙하지 않은 분들께 도움이 되었기를 바랍니다. 감사합니다.