**Embed youtube player in ViSAP**

**Index:**

1. Introduction
2. Iframe API parameters
3. Iframe API events
4. Iframe API methods
5. Embed iframe API in ViSAP
6. Introduction
7. Initiate YouTube player
8. Upload YouTube video ID
9. Add Video title to video menu
10. Create tag and links
11. Tag functionality
12. Create Attributes [ Sketches, Annotation and Question ]
13. Display Attributes [ Sketches, Annotation and Question ]
14. Custom control implementation

**Introduction:**

The IFrame player API lets you embed a YouTube video player on your website and control the player using JavaScript. Unlike the [Flash](https://developers.google.com/youtube/flash_api_reference) and [JavaScript](https://developers.google.com/youtube/js_api_reference) player APIs, which both involve embedding a Flash object on your web page, the IFrame API posts content to an <iframe> tag on your page. This approach provides more flexibility than the previously available APIs since it allows YouTube to serve an HTML5 player rather than a Flash player for mobile devices that do not support Flash

**Refer:** <https://developers.google.com/youtube/iframe_api_reference>

**Iframe API parameters**

This document explains how to embed a YouTube player in your application and also defines the parameters that are available in the YouTube embedded player.

**Refer:** <https://developers.google.com/youtube/player_parameters>

Below are the parameters used in ViSAP

1. 'autoplay': 1
2. 'controls': 0

**Iframe API events**

The API fires events to notify your application of changes to the embedded player. As noted in the previous section, you can subscribe to events by adding an event listener when [constructing the YT.Player object](https://developers.google.com/youtube/iframe_api_reference#Loading_a_Video_Player), and you can also use the [addEventListener](https://developers.google.com/youtube/iframe_api_reference" \l "addEventListener) function.

**Refer:** [https://developers.google.com/youtube/js\_api\_reference#Events](https://developers.google.com/youtube/js_api_reference%23Events)

Below are the event used in ViSAP

1. 'onStateChange'

**Iframe API methods**

Below are the API methods are used in Framework

1. loadVideoById - Load video to YouTube player using video ID
2. getCurrentTime - Get video current play time
3. pauseVideo - Pause video
4. playVideo - Play video
5. sikeTo - Jump video play to given time
6. stopVideo - Stop video playing
7. mute - Mute value
8. unmute - unmute value
9. isMuted - Checking isMuted
10. setVolume - Set volume with given value
11. getVolume - Get current volume value
12. getDuration - Get total video duration

**Embed iframe API in ViSAP**

1. **Introduction:** Currently in ViSAP HTML5 video element is used to play the MP4 video file, and it is not possible to play the YouTube videos in HTML5 player. To play YouTube videos in ViSAP need to embed YouTube API.

Below given iframe API need refer in HTML page, it will create global YT object itself.

<script src="https://www.youtube.com/iframe\_api"></script>

As part of iframe API some of the global objects are added in the framework, all listed below.

// Youtube player support variables

ns.yt = {

player: null,

enabled: false,

videoState: -1

};

1. **‘player’** will contain the YouTube player object
2. **‘enabled’** will set ‘true’ when YouTube video start playing. Will set ‘false’ when playing other videos
3. **‘videoState’** represent the state of the YouTube player.
4. -1 (unstarted)
5. 0 (ended)
6. 1 (playing)
7. 2 (paused)
8. 3 (buffering)
9. 5 (video cued).
10. **Initiate YouTube player:** Method **‘initYTplayer’** is created in visap.js, as mentioned above once the iframe API referred YT global object will create, there is a method called **‘Player’** which is available in iframe API and method is used to create the YouTube player object. Using player object all action can be done on player.

Below is the block of code used to create the YouTube player object

ns.yt.player = new YT.Player('playerYT', {

height: '390',

width: '640',

videoId: videoID,

playerVars: { 'autoplay': 1, 'controls': 0 },

events: {

'onStateChange': function (e) {

ns.yt.videoState = e.data;

},

}

});

**ns.yt.player** – global YouTube player object, control all player functionalities using this object

**height** – height of the video player

**width** – width of the video player

**videoId** - YouTube video id

**playerVars** – default player settings [ autoplay: 1 – video will start play once video id given, controls: 0 – all player controls will get hidden ]

**event** – used to add event listeners, here onStateChange event is used to find the state of the player.

1. **Upload YouTube video ID:** Upload YouTube video details to data.js (it is user specific).

Using ViTag.AddVideo all details will added into data.js.

ViTag.AddVideo([{ t: $("#fileTitle").val(), d: $("#fileDesc").val(), f: ytVideoID, yt: "true" }]);

‘t’: file tile

‘d’: description

‘f’: videoID

‘yt’: indicates that file source is youtube video

1. **Add Video title to video menu:** Once video title added in to data.js it will come in video title list.
2. **Create tag and links:** For creating Tag and links used below API method to get video current time. Based on current time tags will mapped in data.js

ns.yt.player.getCurrentTime();

1. **Tag functionality:** On selecting video title ‘loadVideoById’ API method will call and set YouTube video.

ns.yt.player.loadVideoById(videoID);

1. **Create Attributes [Sketches, Annotation and Question]:** No changes are done for creating video attributes it is as like creating attributes for normal video. But API method ‘getCurrentTime’ is used to get the video current time.
2. **Display Attributes [ Sketches, Annotation and Question ]:** For displaying also no changes are done apart from getting current time.

1. **Custom control implementation:** Method ‘changeYTstate’ added in player.js which is used to change the video state.

**Refere:** “Iframe API methods” on top