***Videomeet Integration***

**Purpose :** The purpose of this document is to give instructions to integrate videomeet into existing system. This instruction manual contains different type of APIs description and integration with collection of multiple parameters.

**Encryption library** : cryptojs.aes.decrypt

*The Encryption Library provides two-way data encryption. it is used for security purpose.*

**CryptoJS** : *CryptoJS is a growing collection of standard and it is used to secure cryptographic algorithms implemented in JavaScript using best practices and patterns. They are fast, and they have a consistent and simple interface.*

**Secret Key :** *In secret key encryption, since the same key is used to encrypt and decrypt, proper safeguarding and distribution of that key is of paramount importance.*

*Secret key for encryption* = **"2e35f242a46d67eeb74aabc37d5e5d05";**

*Secret key which is for encryption and decryption*

*Use this link to download js code*

[***https://videomeet.in/aesjs.zip***](https://videomeet.in/aesjs.zip%20)

***Use this following script for encryption and decryption***

<script src="aes.js"></script>

<script type="text/javascript">

var key = "2e35f242a46d67eeb74aabc37d5e5d05";

var CryptoJSAesJson = {

stringify: function (cipherParams) {

var j = {ct: cipherParams.ciphertext.toString(CryptoJS.enc.Base64)};

if (cipherParams.iv) j.iv = cipherParams.iv.toString();

if (cipherParams.salt) j.s = cipherParams.salt.toString();

return JSON.stringify(j);

},

parse: function (jsonStr) {

var j = JSON.parse(jsonStr);

var cipherParams = CryptoJS.lib.CipherParams.create({ciphertext: CryptoJS.enc.Base64.parse(j.ct)});

if (j.iv) cipherParams.iv = CryptoJS.enc.Hex.parse(j.iv)

if (j.s) cipherParams.salt = CryptoJS.enc.Hex.parse(j.s)

return cipherParams;

}

}

function parseForm(formdata) {

var secret = {};

//var formdata = $('#login\_form').serializeArray();

for(let [name, value] of formdata) {

if(secret[`${name}`]) {

if (!secret[`${name}`].push) {

secret[`${name}`] = [secret[`${name}`]];

}

secret[`${name}`].push(`${value}` || '');

} else {

secret[`${name}`] = `${value}` || '';

}

}

var encrypted = CryptoJS.AES.encrypt(JSON.stringify(secret), key, {format: CryptoJSAesJson}).toString();

encrypted = JSON.parse(encrypted);

encrypted = JSON.stringify(encrypted);

return encrypted;

}

function changeResponse(responseTEXT) {

//console.log(JSON.parse(CryptoJS.AES.decrypt(responseTEXT, key, {format: CryptoJSAesJson}).toString(CryptoJS.enc.Utf8)));

return JSON.parse(CryptoJS.AES.decrypt(responseTEXT, key, {format: CryptoJSAesJson}).toString(CryptoJS.enc.Utf8));

//return eval('(' + responseTEXT + ')');

}

</script>

REQUEST : *This is the request upload video*

Method : POST

Parameters : *Following* ***Parameters*** *used in Upload LOGO API :*

API URL : [api.videomeet.in/v3/fileuploadapi.php](http://10.11.13.153/v3/fileuploadapi.php)
Parameters :

1. filename => .mpt( allowed extension mp4) (Mandatory)

2. formdata => *This is in encrypted format of multiple keys. These keys are explained below*

 {"ct":"B0buPJdNcmB9GFmRCDcIIjpOYVK+PnXWhYshi4K99O2qG1Q+LFDAUdRZpSDkSsuWTMEKooYmU1c4xUW8trwzfB\/l2Nmb50adg9SfPo8mLJqx\/KPXXojD7\/Xy9ZlZkoBP","iv":"a4afb1a7a87929c9e464c96516879c54","s":"e167ecec13a500b0"}

 formdata : (encrption with these keys)

1. authkey => M2atKiuCGKOo9Mj3 (Mandatory)

2. username => sho (Mandatory)

3. password => 1234 (Mandatory)

4. roomname => pcode (Mandatory)

Response :

 **Encrypted :** *Following encrypted response we are receiving this is in encrypted format. We have to decrypt it*

 {"ct":"6Lwrd7d\/cCTvfNams291fRHmZK2c9vGrIJLrzta0kVq2TlXoKJ2oxdjnSdvYvhjPJSt8p+L\/scBj+3ONi7ZTVA==","iv":"f64286f5975fe8312318a91298e4892b","s":"f2518f54e6de5d76"}

 **Decrypted(json)** : *Once we decrypted, we received below response*

 {"status":1,"msg":"File uploaded successfully","data":""}

 {"status":0,"msg":"invalid file type only mp4 allowed.","data":""}

 {"status":0,"msg":"Parameters are missing.","data":""}

 {"status":0,"msg":"Invalid auth key.","data":""}