

API Specification
Version 3.4
support@bopoda.com

Contents

BME API Server API.....	1
Encode Operation.....	1
Cancel Operation API.....	4
Get Operation API.....	4
BME Streaming Server API.....	6
Token API.....	6
Videojs Code Example.....	7

When calling all BME API Server APIs, the Code parameter needs to be added in the request query. An example is shown below:

```
GET {YourAPIURL}?code={BME_API_Server_App_key} HTTP/1.1
```

BME API Server API

Encode Operation

API URL: {BME API Server URL}/EncodeVideo

Method: POST

Body Parameters:

Parameter Name	Type	Required	Description
VideoUrls	List<string>	Y	List of video URLs. Currently, only one video URL is supported, but in the future, support for multiple video URLs will be added. The url recommends using Azure Blob URL. Please note

Parameter Name	Type	Required	Description
			that if the Azure Blob does not allow anonymous access, use a URL with a SAS token and ensure that the token is still valid during the encoding period.
CallbackUrl	string	N	Callback URL for video analysis. This API will be automatically called after encoding is completed. It must be a public POST method. Refer to the Callback parameters section below for the body parameters
CustomData	string	N	Custom business or other information. It will be returned in the callback or when accessing encoding details.
PrivateStreamUrl	bool	N	Whether to enable token authentication for the encoded streaming URL. Default is false. If set to true, the returned HlsUrl and DashUrl cannot be played directly in a browser for live streaming. A query parameter named token must be added to the URL. The value of the token can be obtained through the Token API

Example

POST {BME API Server URL}/EncodeVideo

Body

```
{
  "VideoUrls": [{"your_mp4_url"}],
  "CallbackUrl": {"your_callbakc_url"},
  "CustomData": "Custom message"}

```

Response:

HTTP Status Code 200: Success

Parameter Name	Type	Description
JobId	string	Task ID
videoUrls	List<string>	Video URLs passed during the encoding task
CustomData	string	Custom information passed during the encoding task
CreationTime	datetime	Encoding task creation time
Status	int	Encoding task status, with 0 for in queue, 1 for encoding in progress, 2 for encoding complete, 3 for encoding failure, and 4 for encoding task cancelled
Results	List<Model>	Encoding results, where the format of Model is {"VideoUrl":"","HlsUrl":"","DashUrl":""}
Duration	timespan	Encoding duration
CallbackUrl	string	Callback URL passed during the encoding task

```
{
  "JobId": "",

```

```
"VideoUrls":["{your_mp4_url}"],
"CallbackUrl":"{your_callback_url}",
"CustomData":"Custom message",
"CreateTime":"2023-12-12 00:00:00",
"Status":2,
"Results":[{"VideoUrl":"","HlsUrl":"","DashUrl":""}],
"Duration":"00:20:00"
}
```

Callback parameters

```
{
  "JobId": "",
  "VideoUrls":["{your_mp4_url}"],
  "CustomData":"Custom message",
  "CreateTime":"2023-12-12 00:00:00",
  "Status":2,
  "Results":[{"VideoUrl":"","HlsUrl":"","DashUrl":""}],
  "Duration":"00:20:00"
}
```

Cancel Operation API

API URL: {BME API Server URL}/EncodeVideo/{jobId}/Cancel

Method: Post

Route Parameters:

Parameter Name	Type	Required	Description
jobId	string	Y	Task ID

Example

POST {BME API Server URL}/EncodeVideo/{jobId}/Cancel

Response:

HTTP Status Code 200: Success

Get Operation API

API URL: {BME API Server URL}/EncodeVideo/{jobId}

Method: Get

Route Parameters:

Parameter Name	Type	Required	Description
jobId	string	Y	Task ID

Example

GET {BME API Server URL}/EncodeVideo/{jobId}

Response:

HTTP Status Code 200: Success

Parameter Name	Type	Description
JobId	string	Task ID
videoUrls	List<string>	Video URLs passed during the encoding task
CustomData	string	Custom information passed during the encoding task
CreationTime	datetime	Encoding task creation time
Status	int	Encoding task status, with 0 for in queue, 1 for encoding in progress, 2 for encoding complete, 3 for encoding failure, and 4 for encoding task cancelled
Results	List<Model>	Encoding results, where the format of Model is {"VideoUrl":"","HlsUrl":"","DashUrl":""}
Duration	timespan	Encoding duration

Parameter Name	Type	Description
CallbackUrl	string	Callback URL passed during the encoding task

```
{
  "JobId": "",
  "VideoUrls":["{your_mp4_url}"],
  "CallbackUrl":"{your_callback_url}",
  "CustomData":"Custom message",
  "CreateTime":"2023-12-12 00:00:00",
  "Status":2,
  "Results":[{"VideoUrl":"","HlsUrl":"","DashUrl":""}],
  "Duration":"00:20:00"
}
```

HTTP Status Code 404: job not found

BME Streaming Server API

Token API

API URL: {BME Streaming Server Token API URL}/token

Method: Post

Body Parameters:

Parameter Name	Type	Required	Description
Minutes	int	N	Token valid period in minutes. Default is 120 minutes
Url	string	N	HlsUrl or DashUrl

Example

POST {BME Streaming Server Token API URL}/token

Body

```
{"Minutes":60,"Url":"{your_hlsurl_or_dashurl}"}
```

Response:

HTTP Status Code 200: Success

{your_token}

Videojs Code Example

Below is the html sample code of using videoJS to playback the encoded video url (returned from the Encode API)

(When using the Encode API for encoding, if the privateStreamUrl parameter is set to false, the HlsUrl in the returned result will be public and can be accessed freely.)

```
<video id="my-player" class="video-js"
  controls preload="auto"
  poster="//vjs.zencdn.net/v/oceans.png" data-setup='{}'>
  <source src="https://{BME Streaming Server URL}/output/stream-ba128359-338e-40c5-b687-
d54f66225e9d/hmediaengine.m3u8"
    type="application/x-mpegurl"></source>
  <p class="vjs-no-js">
    To view this video please enable JavaScript, and consider upgrading to a
    web browser that
    <a href="https://videojs.com/html5-video-support/" target="_blank">
      supports HTML5 video
    </a>
  </p>
</video>
```

(When using the Encode API for encoding, if the privateStreamUrl parameter is set to true, the HlsUrl in the returned result will be private, and when accessing it, a query parameter named token must be added. The value of the token can be obtained by calling the Token API.)

```
<video id="my-player" class="video-js"
  controls preload="auto"
  poster="//vjs.zencdn.net/v/oceans.png" data-setup='{}'>
  <source src="https://{BME Streaming Server URL}/output/stream-ba128359-338e-40c5-b687-
d54f66225e9d/hmediaengine.m3u8?token={your token}"
    type="application/x-mpegurl"></source>
  <p class="vjs-no-js">
    To view this video please enable JavaScript, and consider upgrading to a
    web browser that
    <a href="https://videojs.com/html5-video-support/" target="_blank">
```

```
    supports HTML5 video
  </a>
</p>
</video>
```