

Midibus 연동(엔핏)

연동 기본 정보

- API 서버 주소 : <https://api.midibus.kinxcdn.com> (443)
- Player 주소 : play.mbus.tv (80,443)
- Email : enfit01@gmail.com
- User ID : 346
- API Key : 16ade22fbe736695
- 기본 채널 ID : ch_16ad3a60

인증 방식

Basic Auth

- HTTP Basic Auth 표준 인증 방법을 사용
https://en.wikipedia.org/wiki/Basic_access_authentication
- Authentication 헤더
표준 HTTP Authorization 헤더를 사용하여 인증 정보를 전달합니다.

Authorization: Basic *AuthKey*

- AuthKey 생성

AuthKey = Base64(Email+":"+ApiKey)

Token Auth

- Token 요청을 사용하여 발급받은 token을 HTTP 헤더에 포함하여 요청
- Token 헤더 사용
HTTP 헤더를 사용하여 인증 정보를 전달합니다.

X-Mbus-Token: *Token*

연동 Flow

1. Token 발급(Basic Auth)
File Upload에 사용할 Token을 발급 받습니다.
2. File Upload 전송(Token Auth)
파일 업로드를 완료 하면 Media Id가 발급됩니다.
'X-Mbus-Channel' 헤더에 채널 ID를 포함하여 업로드 하면 인코딩이 완료되면 자동으로 배포가 됩니다.
3. Media Status 전송(Token Auth)
파일 업로드로 발급받은 Media Id로 상태를 호출하여 인코딩 결과를 확인합니다.
인코딩이 완료되면 Encoded 값이 true로 응답 되고 Object ID 값이 발급됩니다.(배포 등록 시)
4. Player 주소 사용(인증 불필요)
아래 형식으로 주소를 호출하면 플레이어로 해당 콘텐츠가 재생 됩니다.
<https://play.mbus.tv/ObjectID>

Example

1. Token 생성

Request

```
GET /token?expire=1523602918
```

Response

```
{
  "User": "cdn@kinx.net",
  "Expire": "2018-04-16 16:05:01",
  "Token": "62a0f4f7988d87f583d7b905dcab6d02ad6a30afcc526707587648700f1731b746986AA634DCC69EA21D83DA95FED8F6E2FE04CC6D8"
}
```

2. File 전송

Request

```
POST /v1/upload/4
```

```
X-Mbus-Channel : ch_1c6fabd
```

```
X-Mbus-Token :
```

```
62a0f4f7988d87f583d7b905dcab6d02ad6a30afcc526707587648700f1731b746986AA634DCC69EA21D83DA95FED8F6E2FE04CC6D8
```

Response

```
{
  "success": true,
  "mediald": "16298c8b4c34094b"
}
```

3. Media Status 전송

Request

```
GET /v1/status/16298c8b4c34094b
```

```
X-Mbus-Token :
```

```
62a0f4f7988d87f583d7b905dcab6d02ad6a30afcc526707587648700f1731b746986AA634DCC69EA21D83DA95FED8F6E2FE04CC6D8
```

Response

```
[
  {
    "Mediald": "16298c7b4c34094b",
    "CategoryId": 159,
    "ImgPath": "thumb.midibus.kinxcdn.com/4/16298d7b4c34094b.png",
    "Name": "sample.mp4",
    "Alias": "/sample.mp4",
    "Success": true,
    "Encoded": true,
    "ObjectId": [16227a12cf8a6188],
    "Duration": "00:09:56",
    "Description": null }
]
```

4. Player 주소 사용

직접 Player 호출

```
https://play.mbus.tv/16227a12cf8a6188
```

or

iframe 사용하여 삽입

<iframe width='1920' height='1080' src='https://play.mbus.tv/16227a12cf8a6188' frameborder='0' allowfullscreen></iframe>

Token Create

제한 시간까지 유효한 Token 을 발급합니다.

GET /token

Path Parameter

- expire : 만료 시간(UNIX TimeStamp)

Response : JSON

```
{
  User: 사용자ID,
  Expire : 만료 시간,
  Token : 발급 된 Token
}
```

File Upload

파일 업로드를 수행합니다.

카테고리명을 넣지 않을 경우 미분류 카테고리로 전송

채널명을 넣지 않을 경우 배포되지 않음

POST /v1/upload/[UserID]/[CategoryName]

Path Parameter

- UserId : User ID
- CategoryName : 카테고리명(Optional)

Header Parameter

- X-Mbus-Token : Token
- X-Mbus-Channel : 채널 ID(Optional)
- X-Mbus-Path : 디렉터리 path(CategoryName이 있을 경우 동작)(Optional)

Form Data Parameter

- file : 전송 파일

Response : JSON

```
{
  success : [true|false],
  mediald : 미디어 ID
}
```

GET Media Status

요청하는 Media ID에 해당하는 상태 정보를 전송

GET /v1/status/[MediaId]

Path Parameter

- MediaId : 미디어 ID

Header Parameter

- X-Mbus-Token : Token

Response : JSON

```
[
{
  MediaId : 미디어 ID,
  CategoryId : 카테고리 ID,
  ImgPath : 대표 이미지 경로,
  Name : 미디어 명,
  Alias : 별칭 경로,
  Success : [true|false],
  Encoded : [true|false],
  Object : Object ID List,
  Duration : 영상 재생 시간,
  Description : 사용자 필드
}
]
```

GET Object Status

요청하는 Object ID에 해당하는 정보를 전송

GET /v1/object/[ObjectID]

Path Parameter

- ObjectID : 배포 ID

Response : JSON

```
{
  Objectid : 배포 ID,
  ChannelId : 채널 ID,
  MediaId : 미디어 ID,
  CategoryId : 카테고리 ID,

  Alias : 별칭 경로
}
```

GET Data Transfer Bytes

해당 기간의 Data 사용량 정보 전송(누적)

GET /v1/stat/transfer/[UserID]

Path Parameter

- UserId : 유저 ID

Query Parameter

- token : token
- start : Start Date(yyyyMMddHHmm)
- end : End Date(yyyyMMddHHmm)

Response : JSON

```
{
  User : 사용자 Email,
  Start : Start Date,
  End : End Date,
  TransferBytes : 전송량
}
```

GET Data Transfer Bytes Detail

해당 기간의 Data 사용량 정보 전송(누적)

GET /v1/stat/transfer/[UserID]/detail

Path Parameter

- UserId : 유저 ID

Query Parameter

- token : token
- start : Start Date(yyyyMMddHHmm)
- end : End Date(yyyyMMddHHmm)

Response : JSON

```
{
  User : 사용자 Email,
  Start : Start Date,
  End : End Date,
  Channels : [{ ChannelId : 채널ID, TransferBytes : 전송량 }...]
}
```

GET Storage Usage

해당 기간의 Storage 사용량 정보 전송(최대값)

GET /v1/stat/storage/[UserID]

Path Parameter

- UserId : 유저 ID

Query Parameter

- token : token
- start : Start Date(yyyyMMddHHmm)
- end : End Date(yyyyMMddHHmm)

Response : JSON

```
{
  User : 사용자 Email,
  Start : Start Date,
  End : End Date,
  UsedBytes : 사용량,
  ContentCount : 컨텐츠 수
}
```

DELETE Media

요청하는 Media ID에 해당하는 Media를 삭제

DELETE /v1/category/[CategoryId]/[MediaId]

Path Parameter

- CategoryId : 카테고리 ID
- MediaId : 미디어 ID

Header Parameter

- X-Mbus-Token : Token

Response : message

DELETE Object

요청하는 Object ID에 해당하는 Object의 배포 정보 삭제

DELETE /v1/object/[ObjectId]

Path Parameter

- ObjectId : 오브젝트 ID

Header Parameter

- X-Mbus-Token : Token

Response : message

PUT Media Object

요청하는 Media ID에 해당하는 Media를 채널로 배포

PUT /v1/category/[CategoryId]/[MediaId]

Header Parameter

- X-Mbus-Channel : 채널 ID
- X-Mbus-Token : Token

Path Parameter

- CategoryId : 카테고리 ID
- MediaId : 미디어 ID

Response : JSON

```
{  
  ChannelId : 채널 ID,  
  ObjectId : 오브젝트 ID  
}
```

GET Category List

요청하는 User ID에 해당하는 Category 리스트를 전송

GET /v1/category/[UserId]

Header Parameter

- X-Mbus-Token : Token

Path Parameter

- UserId : 유저 ID

Response : JSON

```
{  
  UserId : 유저 Id,  
  CategoryList : [  
    {  
      CategoryId : 카테고리 ID,  
      CategoryName : 카테고리 명,  
      ChannelId : 자동 배포 채널 ID  
    },  
    ...  
  ]  
}
```

GET Media List

요청하는 Category에 해당하는 Media 리스트를 전송

GET /v1/media/[CategoryId]

Header Parameter

- X-Mbus-Token : Token

Path Parameter

- CategoryId : 카테고리 ID

Query Parameter(Optional)

- nextkey : 이전 요청에서 받은 값 전달
- limit : 최대 리스팅 수(max 1000)
- path : 디렉터리 경로

Response : JSON

```
{
  CategoryId : 카테고리 Id,
  NextKey : 다음 요청 Key(없을 시 -1),
  MediaSimpleInfoList: [
    {
      MediaId : 미디어 ID,
      Name : 미디어 명,
      ThumbPath : 이미지 URL
      State : 상태(1:활성,0:비활성)
      Created : 생성 TimeStamp
      Modified : 수정 TimeStamp
      ObjectIds : 배포 Id(복수 시 ','로 구분)
      Encoded : 인코딩 완료 상태(1:완료, 0:진행)
      Success : 인코딩 성공(1:성공, 0:실패)
    }
    , ...
  ],
  MediaDirList: [
    {
      DirName : 디렉토리명
      DirPath : 디렉토리 경로
      Created : 생성 TimeStamp
    }
    , ...
  ]
}
```

GET User Image Capture

사용자 이미지 추출

GET /v1/capture/[CategoryId]/[MediaId]

Header Parameter

- X-Mbus-Token : Token

Path Parameter

- CategoryId : 카테고리 ID
- MediaId : 미디어 ID

Query Parameter

- timestamp : 추출 위치(ms)
- width : 이미지 가로 크기
- height : 이미지 세로 크기

Response : JSON


```
{  
  Width : 이미지 가로 pixel,  
  Height : 이미지 세로 pixel,  
  ImgPath : 이미지 URL  
  Mediald : 미디어 ID,  
  CategoryId : 카테고리 ID,  
  Timestamps : [ 시간(milliseconds), ... ]  
}
```

example

```
{  
  "Width": 320,  
  "Height": 240,  
  "ImgPath": "thumb.midibus.kinxcdn.com/1/1667ac3e63b57cc4_16672485/1000.jpg",  
  "Mediald": "1667ac3e63b57cc4",  
  "CategoryId": 159,  
  "Timestamp": 1000  
}
```