MCS(Media Control Service) 연동 API

REST-API SPEC

개 정 이 력

변경일	변경내역	변경코드	작성자
2020.03.11	- 최초 작성	1.0	문승완
2020.03.12	모듈 용어 업데이트	1.1	문승완
2020.03.18	API 추가(방송채널 존재여부 / 참여자 확인)	1.2	문승완

※ 본 규격서 내 문의사항 발생 시 작성자 정보를 참조하시기 바랍니다.

목 차

1. 개요

1.1. 목적

이 문서는 COJAM SHOP 서비스에서 WebRTC 미디어 서비스를 제공하기 위해 MCS(Media Control Service) REST-API 연동 인터페이스를 정의하는 문서이다.

1.2. 적용범위

이 문서에서 정의하는 연동 인터페이스는 현재 MCS(Media Control Service) 서버와 COJAMSHOP WAS 간의 연동 인터페이스에 한해서 적용 될 수 있다.

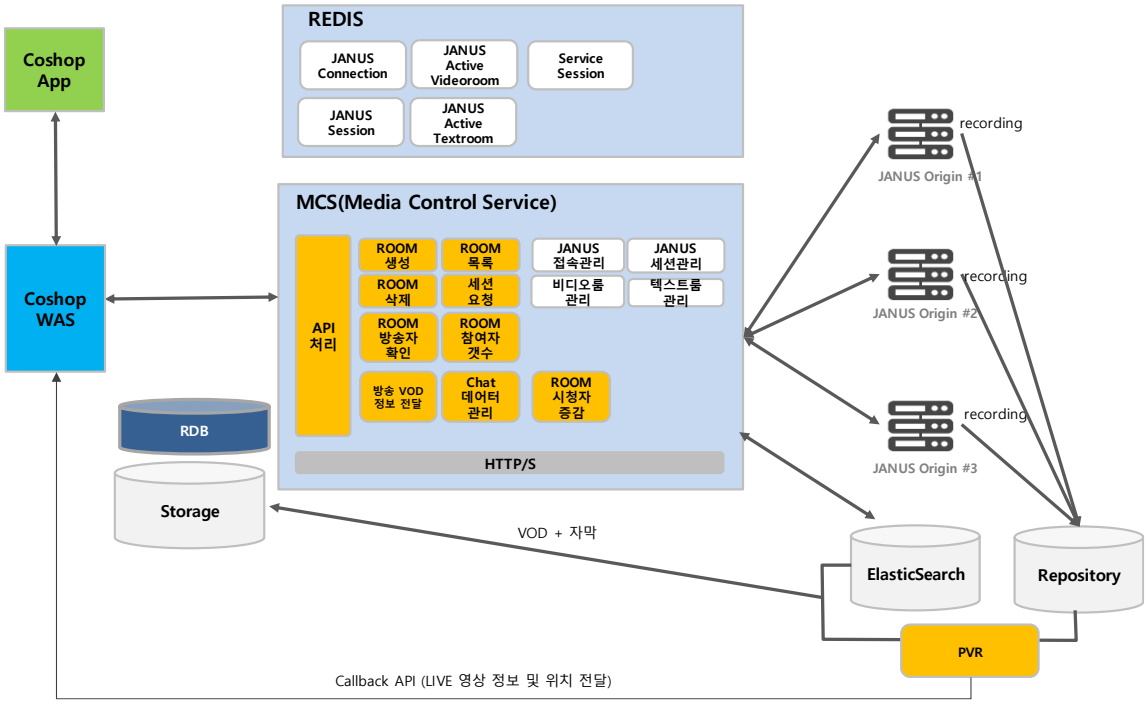
1.3. 약어 및 용어 정리

약어/용어	원어/설명

2. 인터페이스 개요

2.1.1. 시스템 구성

COJAM SHOP 서비스에서 미디어 관련 서비스를 제공하기 위한 핵심 플랫폼인 Media Control Service 의 전체 구성과 COSHOP WAS 에서 연동하기 위한 기본 플로우를 도식화하여 보여준다.



2.2. 연동 시스템 정의

2.2.1. 연동 대상 시스템

2.2.1.1. MCS(Media Control Service)

Coshop WAS 에서 WebRTC 미디어서비스를 이용하기 위해 필요한 API 를 제공하는 핵심 서비스 플랫폼이다.

2.2.1.2. COSHOP WAS

현재 Cojam Shop 서비스를 개발하고 있는 파트너의 웹서버를 말한다.

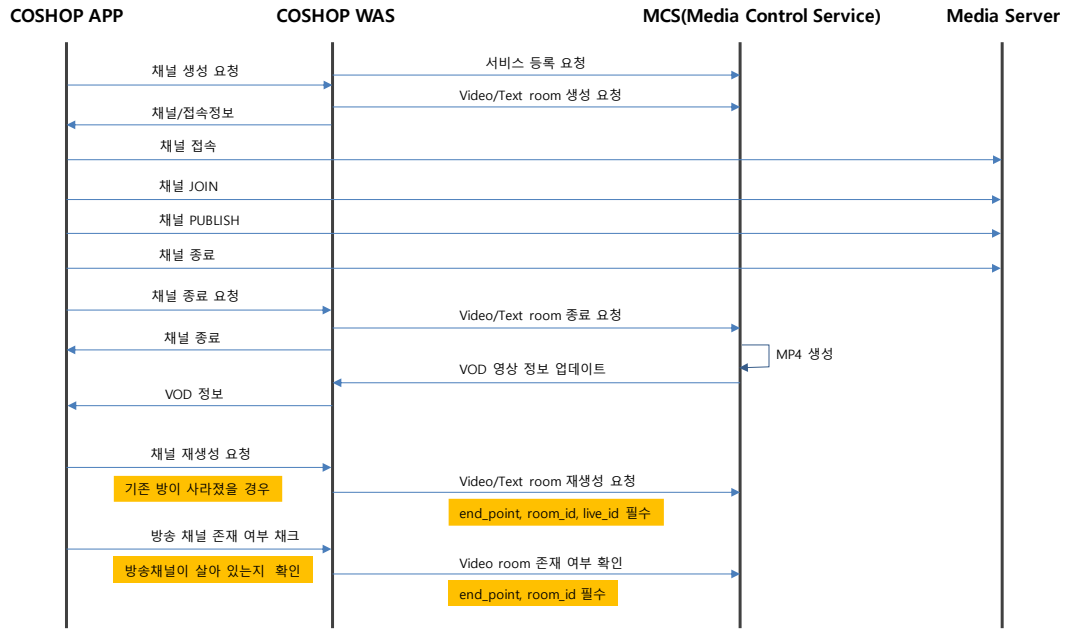
2.2.1.3. COSHOP App

현재 React Native 기반으로 개발 중인 Cojam shop Application 에서 WebRTC 미디어 서비스를 연동한다. WebRTC 미디어 서비스의 손쉬운 연동을 위한 방송자 및 시청자용 React Native 샘플 어플을 제공한다.

2.2.2. 연동 플로우

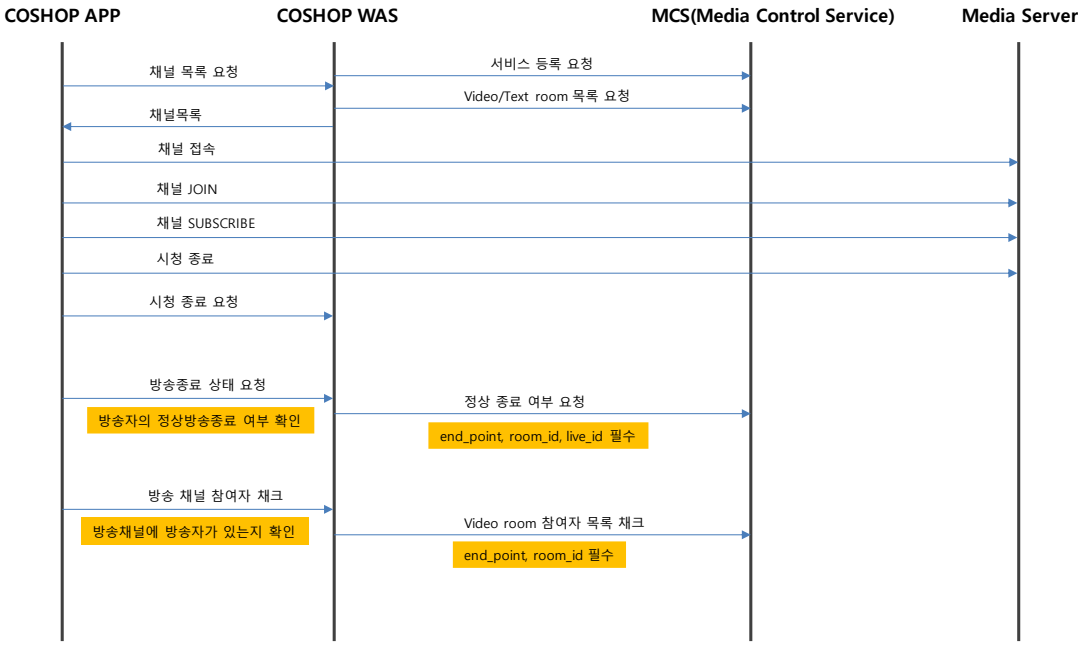
2.2.2.1. 방송자 연동 플로우

방송자 어플에서 WebRTC 미디어 서비스 연동을 위한 Sequence Diagram 을 기술한다.



2.2.2.2. 시청자 연동 플로우

시청자 어플에서 WebRTC 미디어 서비스 연동을 위한 Sequence Diagram 을 기술한다.



2.3. 네트워크 개요

2.3.1. 전송 프로토콜

Layer	Protocol
Application	HTTP
Transport	TCP
Network	IP
Data Link	IEEE802.3/Ethernet
Physical	10/100BaseT

2.3.2. 연동 정보

Server/Client	API 서버 / COSHOP WAS *(테스트 서버 : https://cojam.iptime.org)*
Port	서버 포트: **9100**

HTTP Version	HTTP/1.0, HTTP /1.1
지원 METHOD	POST / GET

2.3.3. 접속 제어

2.3.3.1. 제한사항

- Chunked Stream 방식은 지원하지 않는다.

2.3.3.2. Server

- 다중 요청 시, 요청 정보를 수신한 connection으로 응답하는 것을 원칙으로 하며, 응답 시, 연결된 connection이 해제되었거나 전송 불가능한 상태인 경우에는 Fail 처리한다.

2.3.3.3. Client

- 클라이언트는 요청 시, 연결을 설정하여 요청 메시지를 전송하고, 일정 시간 동안 응답을 대기하여, 응답을 수신하면 연결을 해제한다.
- 클라이언트는 지정된 서버로의 요청 메시지에 대해 응답이 없는 경우, 타임아웃 처리 후에, 자체 정책에 따라 일정 회수만큼 새로운 연결을 통해서 재시도 할 수 있다.

2.3.3.4. Callback

- API서버(WebRTC 미디어 플랫폼)에서 COSHOP WAS 에서 지정한 특정 API를 호출하는 플로우가 포함된다.

2.3.4. 흐름 제어

- 기본적으로 Time-out 또는 비정상 응답 중 재전송이 필요한 경우를 판단하여 재전송을 수행하여야 한다.
- 재전송 알고리즘은 클라이언트의 서비스 중요성 및 서비스 속성에 따라 판단될 수 있다.

2.4. 개발 환경

- 개발 언어: 제약 없음
- 개발 특이 사항: 특이사항 없음

- 개발과 연계된 3rd Party SW: 특이사항 없음

3. 연동 규격

3.1. 개요

3.1.1. 전송 규격

- 데이터 전송은 HTTP 프로토콜을 이용하여 전송한다.
- 메소드는 POST 및 GET을 지원하며 조회 요청 시 사용된다.

Method	기능
POST	등록, 변경 요청 전송에 사용
GET	조회 요청을 위해 사용

3.1.2. 데이터 규격

전송 데이터는 HTTP Body 에 JSON 형식으로 표현된다.

json 기반의 자료형은 json.org 에서 정의하는 다음의 유형을 사용할 수 있다.

Value Type	Description
Object	Object 는 { } 로 정의하며 { String : Value }의 포맷을 가진다.
Array	Array 는 [ ]로 정의하며, [Value, Value, ...] 의 포맷을 가진다.
String	String 은 "" 로 정의하며, 내부에 "와 /를 제외한 어떠한 문자값을 가질수 있다. 특수 문자로 ₩", ₩₩, ₩/ ₩b, ₩f, ₩n, ₩r, ₩t, ₩u(유니코드) 의 값을 사용할 수 있다.
Number	8 진수, 16 진수를 제외한 자연수, 소수, 지수(E +/- 00) 표현 방식을 사용할 수 있다.
Boolean	true, false 로 표현한다.
Null	null 로 표현한다.

** 바이너리 포맷은 byte[]을 BASE64 인코딩 하여 String 타입으로 표현한다.*

3.2. 규격 정의

3.2.1. 전송 규격

3.2.1.1. HTTP Request line 규격

HTTP Request-Line 은 다음과 같이 정의한다. (SP: 공백, CR: Carriage return, LF: Line feed)

- Request-Line = Method SP Request-URI SP HTTP_Version CRLF
- Request-URI = /iot/d/notification, /iot/d/commands, /herit-in/herit-cse/{resource name}

3.2.1.2. HTTP 헤더 규격

HTTP 표준 헤더를 기본으로 하되, 인증을 위해 별도 key 생성규칙과 Basic Authentication 을 정의한다.

헤더 속성	데이터	설명
Content-Type	application/json	JSON 타입의 Body 전송
key	{timestamp}sha256({authid}##{remote_ip}})	1581566580561852817F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF
Authorization	Basic base64({id}:{pwd})	Authorization: Basic Y29zaG9wOjEyMzQ=

위에서 key 를 구성하는 앞단의 timestamp 는 현재의 timestamp 값을 넣어준다. 이는 MCS 에서 일정시간 유효성 판단의 근거로 사용한다.(유효시간 : 2 시간)

3.2.2. 데이터 규격

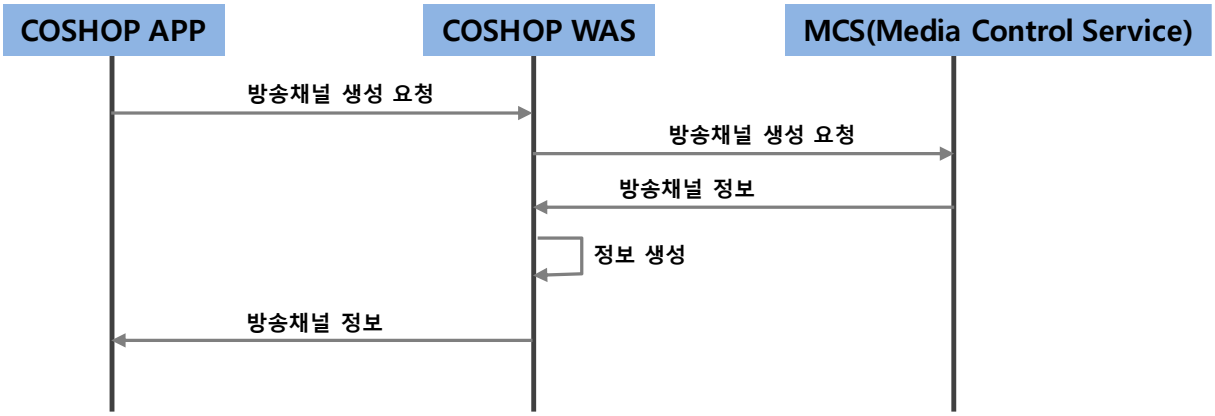
service 별로 JSON 기반의 데이터 표현구조를 정의한다.

3.3. 방송채널 생성 요청 오퍼레이션 정의

3.3.1. 방송채널 생성 요청 처리

COSHOP App 에서 방송채널 생성을 요청할 경우, COSHOP WAS 에서는 MCS 의 API 를 call 하여 방송채널을 생성한다. 이때, COSHOP WAS 는 정상적으로 생성된 방송채널 정보를 WebRTC 클라이언트로 전달한다.

3.3.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.3.1.2. 요청 메시지

HTTP Method	Request URI
POST	/api/v2/create
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름
key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
N/A	
Content	
record	영상 리코딩 여부(true/false)
room_name	방송 채널 이름

description	방송 채널 설명
callback	방송이 완료된 후, 영상이 저장될 위치 저장 위한 coshop was 에서 정의한 API 이름
Example	
POST '{Media Platfom Acess Point}/api/v2/create' \ --header 'key: 1581566580561852817F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \ --header 'Authorization: Basic Y29zaG9wOjEyMzQ=' \ --data-raw '{ "record": true, "room_name": "swmoon1234", "description": "testtesetest", "callback": "http://was.cojam.shop/cb/repo.do" }'	

3.3.1.3. 응답 메시지

Status Code			
200 OK	데이터 요청 처리 성공		
4XX	3.6.2 에러 코드 규격에 따름		
5XX	3.6.3 에러 코드 규격에 따름		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
Content-Length			
Body			
	status	int	상세 결과 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지 정의되지 않았을 경우 빈 문자열
	data	json	Example 참조 access_point : 접속할 미디어 서버 접속정보 live_id : 방송정보 unique id record : 방송 저장 여부

			session_id : 방송세션 id (현재 미사용) textroom_id : 채팅방 ID videoroom_id : 라이브 채널 ID

Example

//성공 시

HTTP/1.1 200 OK

Content-Type: application/json
Content-Length: xxx

{ "status": 200, "message": "success", "data": { "access_point": "wss://cojam.iptime.org:8199", "live_id": "5d6de09f-5520-ef7c-7045-7beb48746094", "record": true, "session_id": 1671309214801837, "textroom_id": 1671309214801837, "textroom_plugin_id": 8880586067021828, "videoroom_id": 1671309214801837, "videoroom_plugin_id": 7813264999024678 } }

//실패 시

HTTP/1.1 500 Internal Error

Content-Type: application/json
Content-Length: xxx

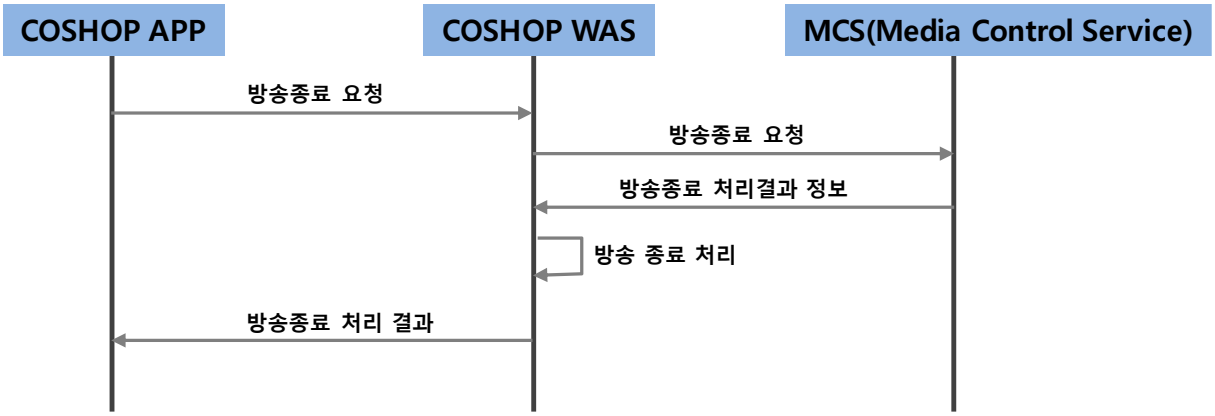
{
"code": "500",
"message": "Database connection error",
"content": {}
}

3.4. 방송 종료 요청 오퍼레이션 정의

3.4.1. 방송 종료 요청 처리

COSHOP APP 에서 COSHOP WAS 로 방송종료 메시지를 받으면, 즉시 MCS(Media Service Control) 로 방송종료 API 를 요청한다.

3.4.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.4.1.2. 요청 메시지

HTTP Method	Request URI
POST	/api/v2/remove/{live_id}
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름
key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
live_id	삭제할 방송채널 UUID
Content	
-	-

Example
POST 'https://cojam.iptime.org:9100/api/v2/remove/a1d7f773-e100-6a97-d8fa-ab1a574502c0' \ --header 'key: 15815665805618528 17F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \ --header 'Authorization: Basic Y29zaG9wOjEyMzQ=' --data-raw '{ "end_point": "wss://cojam.iptime.org:8199", "room_id": "12124324343434", "live_id": "5d6de09f-5520-ef7c-7045-7beb48746094" }'

3.4.1.3. 응답 메시지

Status Code			
200 OK	데이터 요청 처리 성공		
4XX	3.6.2 에러 코드 규격에 따름		
5XX	3.6.3 에러 코드 규격에 따름		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
Content-Length			
Content			
오류 정보	code	String	상세 오류 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지 정의되지 않았을 경우 빈 문자열

Example
//성공 시 HTTP/1.1 200 OK Content-Type: application/json Content-Length: xxx { "status": 200,

```
   "message": "success"
}
```

//실패 시

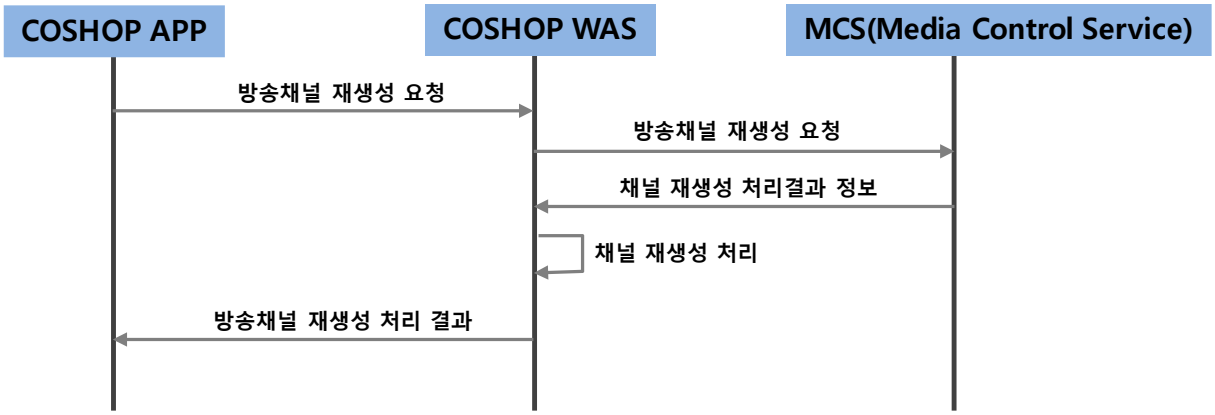
```
HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx

{
      "code": "500",
      "message": "Database connection error",
      "content": {}
}
```

3.5. 방송채널 재생성 요청 오퍼레이션 정의

3.5.1. 방송채널 재생성 요청 처리

방송이 비정상 종료되고, 방송채널이 삭제되었을 경우, COSHOP APP 에서 COSHOP WAS 로 방송채널 재생성 요청을 보내고, COSHOP WAS 는 MCS(Media Control Service)에 방송채널 재생성 요청 메시지를 보낸다.

3.5.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.5.1.2. 요청 메시지

HTTP Method	Request URI		
POST	/api/v2/recreate		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
key	3.2.1.2 HTTP 헤더 규격에 따름		
Authorization	3.2.1.2 HTTP 헤더 규격에 따름		
Parameter			
N/A			
Content			
end_point	미디어서버 접속정보		
room_id	방송채널 ID		
live_id	방송채널 UUID		
Example			
POST 'https://cojam.iptime.org:9100/api/v2/recreate' \ --header 'key: 1581566580561852817F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \ --header 'Authorization: Basic Y29zaG9wOjEyMzQ='			

3.5.1.3. 응답 메시지

Status Code			
200 OK	데이터 요청 처리 성공		
4XX	3.6.2 에러 코드 규격에 따름		
5XX	3.6.3 에러 코드 규격에 따름		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
Content-Length			
Body			
	status	int	상세 결과 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지

			정의되지 않았을 경우 빈 문자열
	data	json	Example 참조 access_point : 접속할 미디어 서버 접속정보 live_id : 방송정보 unique id record : 방송 저장 여부 session_id : 방송세션 id (현재 미사용) textroom_id : 채팅방 ID videoroom_id : 라이브 채널 ID

Example

//성공 시

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: xxx

{ "status": 200, "message": "success", "data": { "access_point": "wss://cojam.iptime.org:8199", "live_id": "5d6de09f-5520-ef7c-7045-7beb48746094", "record": true, "session_id": 1671309214801837, "textroom_id": 1671309214801837, "textroom_plugin_id": 8880586067021828, "videoroom_id": 1671309214801837, "videoroom_plugin_id": 7813264999024678 } }

//실패 시

HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx

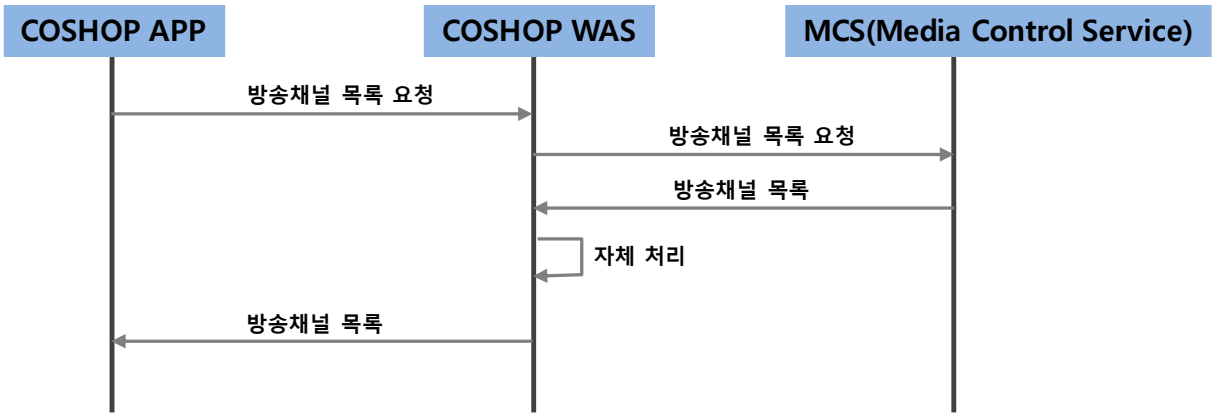
{
"code": "500",
"message": "Database connection error",
"content": {}
}

3.6. 방송채널 목록 전달 오퍼레이션 정의

3.6.1. 방송채널 목록 전달 처리

COSHOP APP(시청자)가 방송채널 목록 요청 메시지를 COSHOP WAS 로 보내면, COSHOP WAS 는 MCS(Media Control Service)으로 방송채널 목록 요청 API 를 call 한다.

3.6.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.6.1.2. 요청 메시지

HTTP Method	**Request URI**
POST	/api/v2/list
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름
key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
N/A	
Content	
-	-
Example	
POST 'https://cojam.iptime.org:9100/api/v2/list' \ --header 'key: 158156658056185281 7F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \ --header 'Authorization: Basic Y29zaG9wOjEyMzQ='	

3.6.1.3. 응답 메시지

<table>
<tr><th colspan="4">Status Code</th></tr>
<tr><td>200 OK</td><td colspan="3">데이터 요청 처리 성공</td></tr>
<tr><td>4XX</td><td colspan="3">3.6.2 에러 코드 규격에 따름</td></tr>
<tr><td>5XX</td><td colspan="3">3.6.3 에러 코드 규격에 따름</td></tr>
<tr><th colspan="4">Header</th></tr>
<tr><td>Content-Type</td><td colspan="3">3.2.1.2 HTTP 헤더 규격에 따름</td></tr>
<tr><td>Content-Length</td><td colspan="3"></td></tr>
<tr><th colspan="4">Body</th></tr>
<tr><td rowspan="3"></td><td>status</td><td>int</td><td>상세 결과 코드
정의되어 있지 않을 경우 HTTP Status Code 와 동일값</td></tr>
<tr><td>message</td><td>String</td><td>오류 메시지
정의되지 않았을 경우 빈 문자열</td></tr>
<tr><td>data</td><td>json</td><td>Example 참조
room_count: 방송채널 개수
room_list : 방송채널 상세목록
live_seq: 방송채널 UUID
room_name: 방송채널 이름
access_point : 접속할 미디어 서버 접속정보
record : 방송 저장 여부
textroom : 채팅방 ID
videoroom : 라이브 채널 ID
description: 방송채널 설명</td></tr>
<tr><th colspan="4">Example</th></tr>
<tr><td colspan="4">//성공 시
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: xxx

{ "status": 200, "message": "success", "data": { "room_count": 1, "room_list": [{ "live_seq": "aa8463a1-da40-ea17-a25d-c05ca7a0926d", "room_name": "swmoon12345678", "access_point": "wss://cojam.iptime.org:8199", "videoroom": "5121700499680100", "textroom": "5121700499680100", "record": true, "description": "testtesetest" }] } }

//실패 시
HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx</td></tr>
</table>

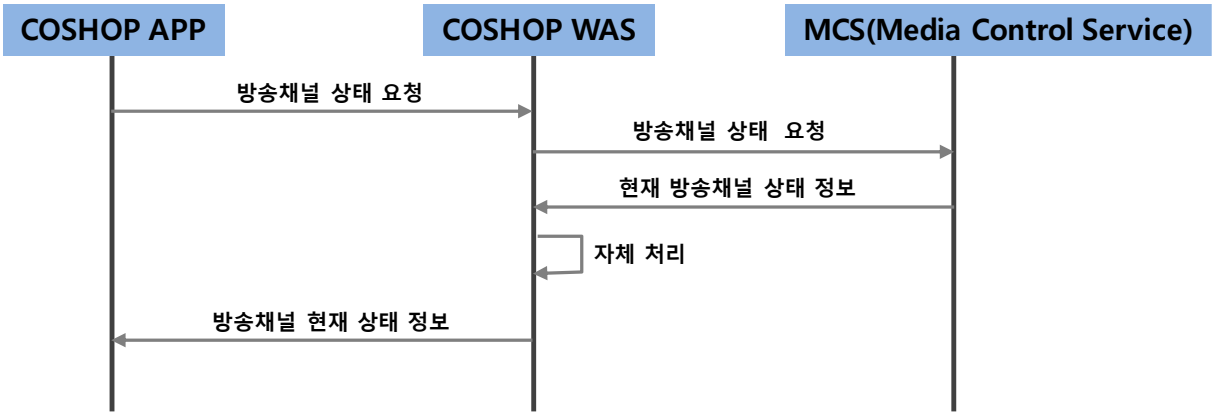
```
{
	"code": "500",
	"message": "Database connection error",
	"content": {}
}
```

3.7. 방송채널 상태 요청 오퍼레이션 정의

3.7.1. 방송채널 상태 요청 처리

COSHOP APP(시청자)에서 현재 방송이 정상적으로 종료되었는지 확인하기 위해 COSHOP WAS 로 상태 요청을 하면, COSHOP WAS 는 MCS(Media Control Service)로 방송채널 상태요청 API 를 call 하여 확인한다.

3.7.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.7.1.2. 요청 메시지

HTTP Method	Request URI
POST	/api/v2/status
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름

key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
N/A	
Content	
end_point	미디어서버 접속정보
room_id	방송채널 ID
live_id	방송채널 UUID
Example	

```
POST 'https://cojam.iptime.org:9100/api/v2/status' \
--header 'key: 158156658056185281 7F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \
--data-raw '{
    "end_point": "wss://cojam.iptime.org:8199",
    "room_id": "5289393826123323",
    "live_id": "757e2f05-4eea-7b17-0092-7a539fb03cf2"
}'
```

3.7.1.3. 응답 메시지

Status Code			
200 OK	데이터 요청 처리 성공		
4XX	3.6.2 에러 코드 규격에 따름		
5XX	3.6.3 에러 코드 규격에 따름		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
Content-Length			
Body			
	status	int	상세 결과 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지 정의되지 않았을 경우 빈 문자열
	data	json	Example 참조

			access_point : 접속할 미디어 서버 접속정보 room_id : 방송채널 ID status : 상태코드 desc: 상태코드 설명 (0000 : 비정상종료, 1000: 방송채널 정상 생성, 9000: 정상 종료)

Example

//성공 시

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: xxx


```
{
    "status": 200,
    "message": "success",
    "data": {
        "access_point": "wss://cojam.iptime.org:8199",
        "desc": "Ended normally",
        "room_id": "5289393826123323",
        "status": "9000"
    }
}
```


//실패 시

HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx


```
{
      "code": "500",
      "message": "Database connection error",
      "content": {}
}
```

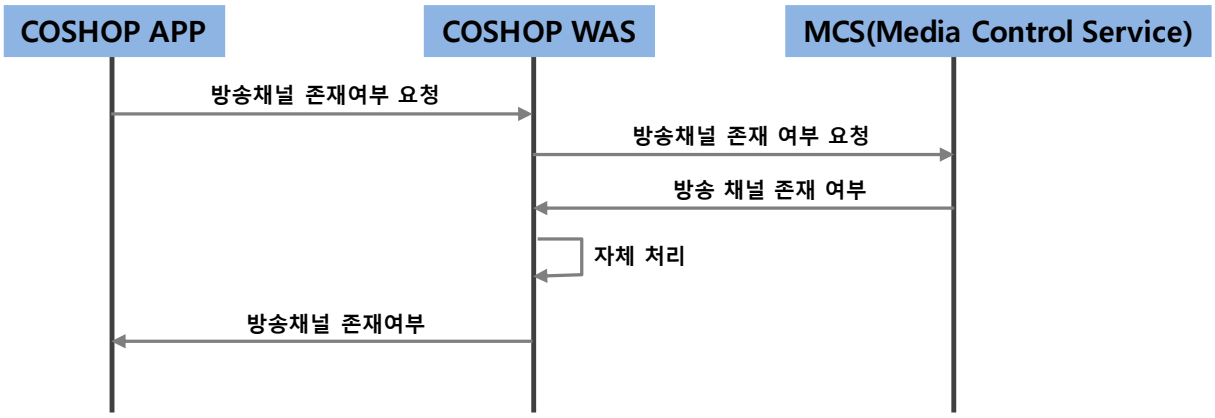

3.8. 방송채널 존재여부 요청 오퍼레이션 정의

3.8.1. 방송채널 존재 여부 요청 처리

COSHOP APP(방송자)에서 재접속 등으로 현재 방송 채널이 아직 살아 있는지 확인하기 위해

COSHOP WAS 로 요청을 하면, COSHOP WAS 는 MCS(Media Control Service)로 방송채널 존재여부 요청 API 를 call 하여 확인한다.

3.8.1.1. 기능 흐름도



Source	Destination
COSHOP WAS	MCS(Media Control Service)

3.8.1.2. 요청 메시지

HTTP Method	Request URI
POST	/api/v2/exist
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름
key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
N/A	
Content	
end_point	미디어서버 접속정보
room_id	방송채널 ID
Example	
POST 'https://cojam.iptime.org:9100/api/v2/exist' \ --header 'key: 1581566580561852817F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \	

```
--header 'Authorization: Basic Y29zaG9wOjEyMzQ=' \
--data-raw '{
        "end_point": "wss://cojam.iptime.org:8199",
        "room_id": "7658049858508502"
}'
```

3.8.1.3. 응답 메시지

Status Code			
200 OK	데이터 요청 처리 성공		
4XX	3.6.2 에러 코드 규격에 따름		
5XX	3.6.3 에러 코드 규격에 따름		
Header			
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름		
Content-Length			
Body			
	status	int	상세 결과 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지 정의되지 않았을 경우 빈 문자열
	data	json	Example 참조 access_point : 접속할 미디어 서버 접속정보 room_id : 방송채널 ID exist : 존재여부

Example

//성공 시

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: xxx

```
{
    "status": 200,
    "message": "success",
    "data": {
        "access_point": "wss://cojam.iptime.org:8199",
        "exists": true,
        "room_id": "7658049858508502"
    }
```

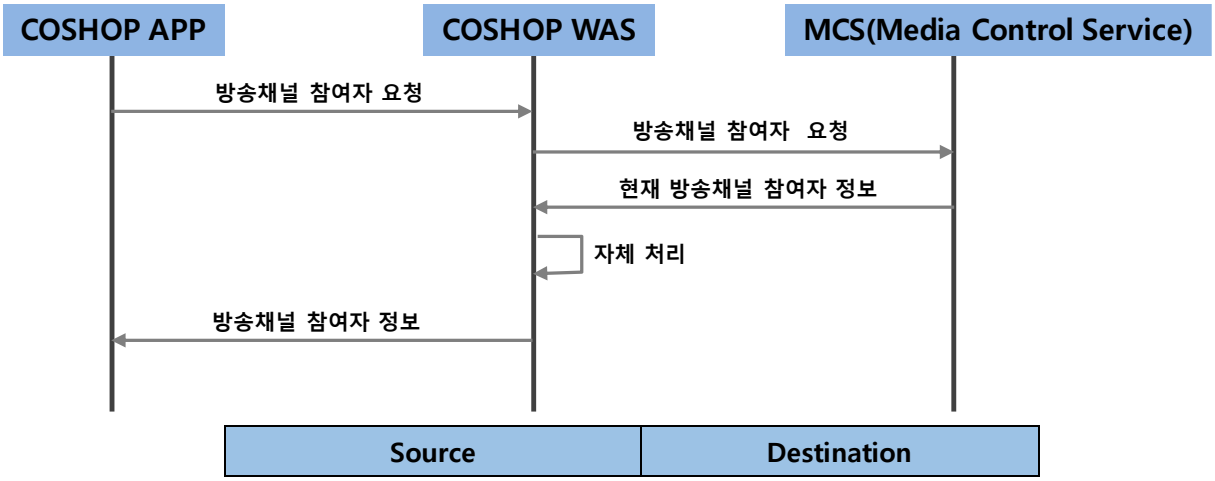
```
}


//실패 시
HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx

{
      "code": "500",
      "message": "Database connection error",
      "content": {}
}
```

3.9. 방송채널 참여자 확인 요청 오퍼레이션 정의

3.9.1. 방송채널 참여자 확인 여부 요청 처리

COSHOP APP(시청자)에서 방송 요청시 현재 방송 채널에서 방송이 되고 있는지 확인하기 위해 COSHOP WAS 로 요청을 하면, COSHOP WAS 는 MCS(Media Control Service)로 방송채널 방송 진행여부(참여자 확인) 요청 API 를 call 하여 확인한다.

3.9.1.1. 기능 흐름도



COSHOP WAS	MCS(Media Control Service)

3.9.1.2. 요청 메시지

HTTP Method	**Request URI**
POST	/api/v2/participants
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름
key	3.2.1.2 HTTP 헤더 규격에 따름
Authorization	3.2.1.2 HTTP 헤더 규격에 따름
Parameter	
N/A	
Content	
end_point	미디어서버 접속정보
room_id	방송채널 ID
Example	

```
POST 'https://cojam.iptime.org:9100/api/v2/participants' \
--header 'key: 158156658056185281 7F38D82AC8EBF84B9E1864FA5C79C006E3EF3C9E2936CA94BE3CBABF ' \
--data-raw '{
    "end_point": "wss://cojam.iptime.org:8199",
    "room_id": "7658049858508502"
}'
```

3.9.1.3. 응답 메시지

Status Code	
200 OK	데이터 요청 처리 성공
4XX	3.6.2 에러 코드 규격에 따름
5XX	3.6.3 에러 코드 규격에 따름
Header	
Content-Type	3.2.1.2 HTTP 헤더 규격에 따름

Content-Length			
Body			
	status	int	상세 결과 코드 정의되어 있지 않을 경우 HTTP Status Code 와 동일값
	message	String	오류 메시지 정의되지 않았을 경우 빈 문자열
	data	json	Example 참조 access_point : 접속할 미디어 서버 접속정보 room_id : 방송채널 ID participants : 방송자(참여자) 아이디 목록 participants_count : 방송자 목록 개수(방송하는 경우 1, 아닌경우 0)

Example

//성공 시

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: xxx

{
    "status": 200,
    "message": "success",
    "data": {
        "access_point": "wss://cojam.iptime.org:8199",
        "participants": [],
        "participants_count": 0,
        "room_id": "7658049858508502"
    }
}
```

//실패 시

```
HTTP/1.1 500 Internal Error
Content-Type: application/json
Content-Length: xxx

{
      "code": "500",
      "message": "Database connection error",
      "content": {}
}
```