

화상 PT, Class 연계 통합 서비스



서울 2반 4조 | 서요셉 곽호근 서민영 안대현 윤영은

CONTENTS

- 01** Team BTS 소개
- 02** 아이디어 및 프로젝트 설명
- 03** 공통 진행 사항
- 04** Part별 진행 사항

01 Team BTS 소개

01 Team BTS 소개

Team BTS Body Training Squad



공정왕 민영

WebRTC



이두박 호근

WebRTC



요세푸스 요셉

Frontend & Backend



발라드킹 대현

Backend



z | 존 영은

Frontend

02

아이디어 및 프로젝트 설명

아이디어 1. Problem

아이디어 2. Insight

아이디어 3. Conclusion

P:T 프로젝트를 소개합니다.

02 아이디어 및 프로젝트 설명

아이디어 1. Problem



&



Home Training

Untact Consumption

운동에 대한 관심이 증가했지만, 감염 우려로 인해 개인 운동과 홈트레이닝 증가

02 아이디어 및 프로젝트 설명

아이디어 2. Insight

집에서도 안전하게
전문가의 코칭, 피드백을 받을 수 없을까?

#바른자세

#운동루틴

#동기부여

#파로또같이

#언택트클래스

02 아이디어 및 프로젝트 설명

아이디어 3. Conclusion



그렇다면… 화상 PT와 Class 신청/수강을 연계한 통합 서비스를 만들어보자!

02 아이디어 및 프로젝트 설명

P:T 프로젝트를 소개합니다.



당신의 **피땀눈물** 헛되지 않도록

화상 수업

1:1 / 1:M 수업으로 혼자서도 여럿이도 즐겁게!

영상 피드백

녹화 영상으로 자세한 자세 교정 및 피드백으로 바르게!

다양한 카테고리

요가, 헬스, 다이어트 등 다양한 운동 종목으로 새롭게!

03 공통 진행 사항

설계 - Wireframe, DB ERD

형상 및 일정관리 - Git, Jira, Notion

03 공통 진행 사항

설계 - Wireframe

메인페이지

MainPage

MainPage(로그인 후)

MainPage(로그인 후)

MainPage(검색 결과 페이지)

MainPage(모달 창-회원가입)

MainPage(모달 창-로그인)

Footer

03 공통 진행 사항

설계 - Wireframe

클래스 상세 페이지



이벤트 페이지



03 공통 진행 사항

설계 - Wireframe

마이페이지 - 수강생(1)

수강생-마이페이지(수강중클래스, MyClass)

수강생-마이페이지(내 정보)

마이페이지(녹화된 영상)

마이페이지(녹화된 영상)

03 공통 진행 사항

설계 - Wireframe

마이페이지(찜한 클래스)

마이페이지(수강원료 클래스)

마이페이지(녹화된 영상 개별 확인 - 피드백 미...)

마이페이지 - 수강생(2)

The wireframe displays the MyPage interface for a student. It includes a header with navigation links, a sidebar with user information, and a main content area. The main content area shows two lists of courses: '찜한 클래스' (Saved Classes) and '수강원료 클래스' (Course Materials). Each list contains course details like name, teacher, and schedule. Below these lists is a video player for a recorded video, with a feedback section below it.

마이페이지(녹화된 영상 개별 확인 - 피드백 완...)

This wireframe shows the video playback screen from the previous one, with a message indicating that feedback has been completed. The video player interface remains the same, showing the recorded video and its controls.

03 공통 진행 사항

설계 - Wireframe

마이페이지 - 트레이너(1)

마이페이지(수강중클래스, MyClass)

트레이너-마이페이지(내 정보)

트레이너-마이페이지(클래스 개설)

트레이너-마이페이지(트레이너 설명)

The wireframes illustrate the design of a Trainer-Mypage application. Each prototype includes a header with tabs for Home, MyPage, MyClass, and MyLesson, a sidebar with '내 정보' (My Info) and '내 클래스' (My Classes) sections, and a main content area.

- 마이페이지 - 트레이너(1):** Shows a list of classes (MyClass) with columns for Name, Description, and Status.
- 마이페이지(수강중클래스, MyClass):** Shows a list of classes with columns for Name, Description, and Status.
- 트레이너-마이페이지(내 정보):** Shows user profile information including a profile picture, name, email, and phone number.
- 트레이너-마이페이지(클래스 개설):** Shows fields for creating a new class, including class name, description, and schedule.
- 트레이너-마이페이지(트레이너 설명):** Shows a detailed profile of a trainer with a large profile picture and descriptive text.

03 공통 진행 사항

설계 - Wireframe

마이페이지 - 트레이너(2)

The image displays three wireframe prototypes for a 'MyPage Trainer (2)' interface, arranged vertically. Each prototype is a rectangular frame with a header, sidebar, and main content area.

- 마이페이지(피드백 등록)**: This prototype shows a feedback registration form. The main content area contains a table with columns for '제작자' (Creator), '제작일' (Creation Date), and '제작내용' (Content). The table has four rows with sample data: '김민경', '2023-01-01', '제작자 A'; '김민경', '2023-01-01', '제작자 B'; '김민경', '2023-01-01', '제작자 C'; and '김민경', '2023-01-01', '제작자 D'. Below the table is a large input field labeled '제작내용' with placeholder text: '제작내용을 등록해주세요.' (Please register the content.)
- 마이페이지(수강완료 클래스)**: This prototype shows a list of completed classes. The main content area contains a table with columns for '제작자' (Creator), '제작일' (Creation Date), and '제작내용' (Content). The table has four rows with sample data: '김민경', '2023-01-01', '제작자 A'; '김민경', '2023-01-01', '제작자 B'; '김민경', '2023-01-01', '제작자 C'; and '김민경', '2023-01-01', '제작자 D'. Below the table is a large input field labeled '제작내용' with placeholder text: '제작내용을 등록해주세요.' (Please register the content.)
- 마이페이지(피드백 등록)**: This prototype shows a feedback registration form. The main content area contains a table with columns for '제작자' (Creator), '제작일' (Creation Date), and '제작내용' (Content). The table has four rows with sample data: '김민경', '2023-01-01', '제작자 A'; '김민경', '2023-01-01', '제작자 B'; '김민경', '2023-01-01', '제작자 C'; and '김민경', '2023-01-01', '제작자 D'. Below the table is a large input field labeled '제작내용' with placeholder text: '제작내용을 등록해주세요.' (Please register the content.)

03 공통 진행 사항

설계 - Wireframe

PT 받기 - 1:1



PT 받기 - 1:N



* PT 받기는 WebRTC를 활용한 화상 수업 화면입니다.

관리자 페이지

관리자-관리탭(요청된 클래스)



관리자-관리탭(이벤트)



관리자-관리탭(이벤트 작성)



관리자-요청된 클래스 확인 모달



관리자-관리탭(이벤트 디테일)



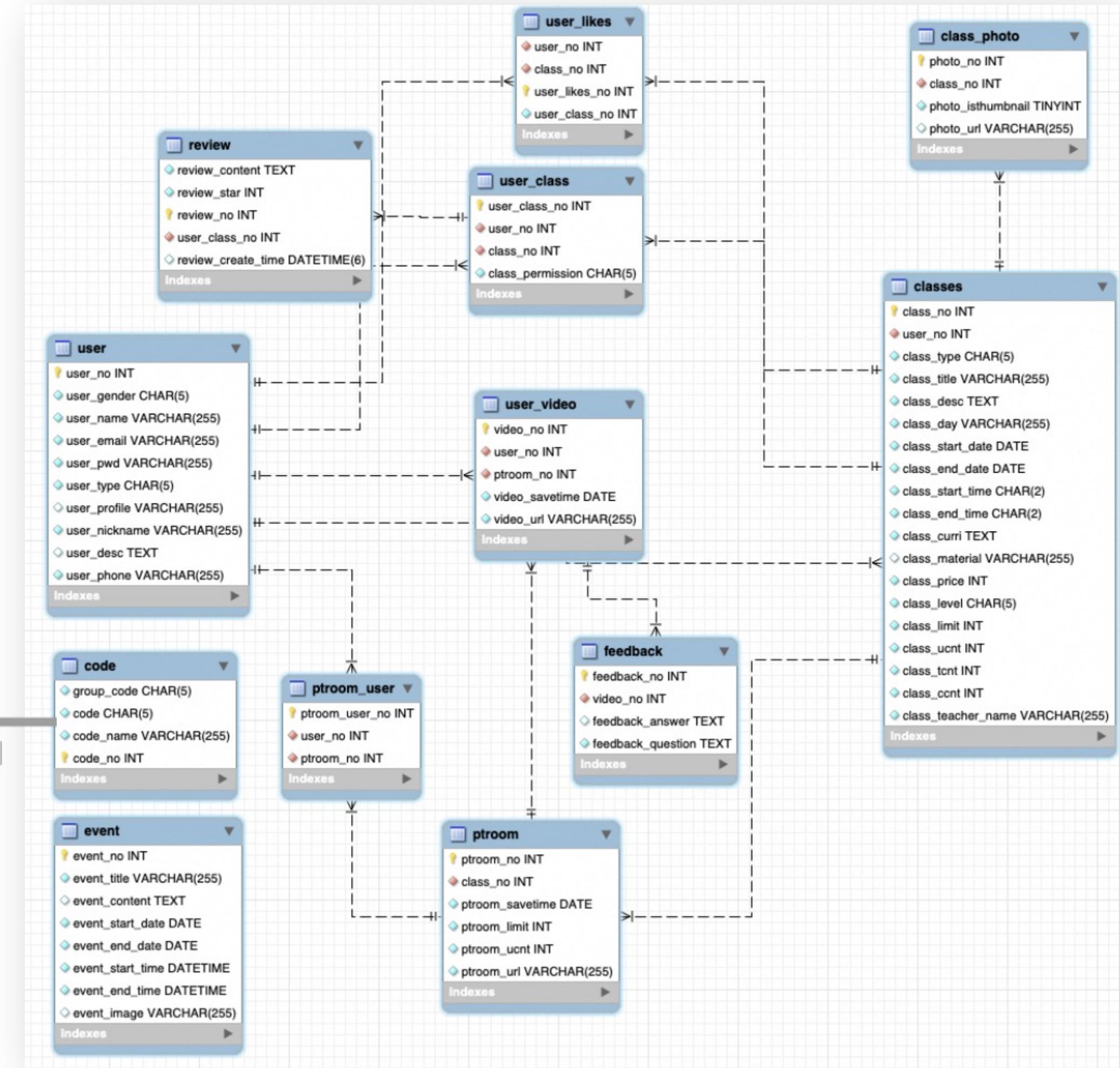
03 공통 진행 사항

설계 - DB ERD

성별, 사용자 구분, 운동 카테고리, 나이도,
클래스 개설 요청 결과를 공통 Code로 관리하여
유지보수에 용이하도록 설계

group_code	code	code_name	code_no
001	001	남자	1
001	002	여자	2
001	003	기타	3
002	001	관리자	4
002	002	트레이너	5
002	003	수강생	6
003	001	헬스	7
003	002	요가	8
003	003	필라테스	9
004	001	입문	10
004	002	초급	11
004	003	중급	12
004	004	고급	13
005	001	승인	14
005	002	미승인	15
005	003	거절	16

공통 Code Table 상태 보기



03 공통 진행 사항

형상 및 일정관리 - Git

1. Git Convention을 정하여 상황별, 파트별 Commit 진행
2. Gitmoji를 활용해 한눈에 작업 분류 파악 가능

3. Git Convention

Git commit message 컨벤션 : " '깃모지' - [작업한파트] 작업내용 "

깃모지	Aa 이름	커밋 메시지
🎉	init	:tada:
✨	기능 추가	:sparkles:
♻️	기능 수정	:recycle:
⚡	성능 개선	:zap:
🐛	버그 수정	:bug:
⚰️	코드 삭제	:coffin:
📱	반응형 디자인 작업	:iphone:
🙈	깃 ignore	:see_no_evil:
📝	문서 작성 (README 등)	:memo:
👽	나땅베러~~than 유우우우 (기타 커밋)	:alien:

+ 새로 만들기

계수 10

s05-webmobile1-sub2 / PiT · C · lab.ssafty.com/s05-webmobile1-sub2/S05P12A204

Projects Groups More

PiT Project ID: 55051 Leave project

97 Commits 5 Branches 0 Tags 179 MB Files 179 MB Storage

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the Auto DevOps documentation

Enable in settings

master S05P12A204 / + History Find file Web IDE Clone

[BE] - devtools 추가 yoseph0310 authored 2 hours ago d1690edc

Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster

Set up CI/CD

Name	Last commit	Last update
backend	:sparkles: [BE] - devtools 추가	2 hours ago
frontend	:sparkles: [FE] update main-header.vue	2 hours ago
.gitignore	:recycle: [master] - update gitignore	20 hours ago

03 공통 진행 사항

형상 및 일정관리 - Jira

1. 명명 규칙에 따른 Issue 등록: [Part] - Issue Description
2. Jira-Mattermost 연동으로 빠른 변동사항 확인
3. 보고자, 담당자, 시간을 자세히 작성하여 효과적인 분업

A screenshot of the Mattermost messaging application. The channel is named 'A204팀'. Recent messages include:

- jira: 오전 5:54 운영은 updated Sprint from "None" to "개발 1주차" on 이야기 S05P12A204-79: [FE] - 마이페이지
- jira: 오전 5:54 운영은 updated Rank from "none" to "ranked higher" on 이야기 S05P12A204-79: [FE] - 마이페이지
- jira: 오후 8:52 서요셉 updated status from "In Progress" to "In Progress" on 이야기 S05P12A204-77: [BE] - webpack 관련
- jira: 오전 12:13 서요셉 commented on 이야기 S05P12A204-77: [BE] - webpack 관련
 - frontend에서 npm run build 가 먼저 되어야함 dist 폴더로 build 까지는 됨. 하지만 sts에서 ?로 뜯 인식을 못하는 듯 SecurityConfig 에서 antMatcher도 하용. -> 안됨 WebConfig에서 registry, application.properties에서 정적 파일 저장소 dist로 매핑 -> 안됨
- jira: 오전 12:57 서민영 updated Story Points from "None" to "40" on 이야기 S05P12A204-84: [WR] - EC2 instance에 Kurento server 등록
- jira: 오전 12:57 서민영 assigned 서민영 to 이야기 S05P12A204-84: [WR] - EC2 instance에 Kurento server 등록
- jira: 오전 1:01 서민영 assigned 곽호근 to 이야기 S05P12A204-90: [WR] - AWS에서 Kurento 실습
- jira: 오전 1:03 서민영 updated Story Points from "None" to "20" on 이야기 S05P12A204-90: [WR] - AWS에서 Kurento 실습

The bottom of the screen shows a text input field: 'A204팀에 글쓰기'.

A screenshot of the Jira Software interface. The board is titled 'S05P12A204 보드' and has a section titled '백로그' (Backlog). The backlog list shows the following issues:

- 환경설정 및 기본 기능 개발 23 이슈 활성
 - S05P12A204-13 [WR] - 개발환경 설정 (WebRTC, 12)
 - S05P12A204-20 [WR] - 예제 코드 실습 (WebRTC, 28)
 - S05P12A204-17 [FE] - 메인페이지 (FrontEnd, 12)
 - S05P12A204-19 [FE] - 클래스 상세페이지 (FrontEnd, 16)
 - S05P12A204-50 [BE] - API url 정리 (BackEnd, 2)
 - S05P12A204-53 [BE] - AWS 서버 / DB 등록 (BackEnd, 10)
 - S05P12A204-54 [BE] - AWS RDS & WorkBench Failed Connection (BackEnd, 0)
 - S05P12A204-30 [BE] - DB 테이블 작성 (BackEnd, 3)
 - S05P12A204-35 [BE] - 회원가입 (BackEnd, 8)
 - S05P12A204-41 [BE] - JWT Token 기능 (BackEnd, 16)
 - S05P12A204-48 [BE] - 메인페이지 클래스 목록 (BackEnd, 2)
 - S05P12A204-49 [BE] - 클래스 상세페이지 구현 (BackEnd, 4)
 - S05P12A204-39 [BE] - 로그인 (BackEnd, 4)
 - S05P12A204-45 [BE] - 회원 정보 조회/수정/탈퇴 (BackEnd, 1)

On the right side, there is a detailed view of the first issue, 'PiT / S05P12A204-13':

- [WR] - 개발환경 설정
- 예상: 12
- 자세히:
 - 상태: 완료 (작업 흐름 보기)
 - 우선순위: Medium
 - 구성 요소: 없음
 - 레이블: 없음
 - 영향 받는 버전: 없음
 - 수정 버전: 없음
 - Epic Link: WebRTC
- 사람:
 - 보고자: 곽호근
 - 담당자: 곽호근
 - 나에게 할당: 나에게 할당
- 날짜:
 - 생성일: 21/7/21 2:27 오후
 - 변경일: 어제 10:04 오전
- 설명

03 공통 진행 사항

형상 및 일정관리

- Notion

1. Calander 활용을 통한 회의록 및 컨설턴트 미팅 정리
2. 팀 내 정책과 코딩 컨벤션과 같은 중요사항 공유
3. 프로젝트 진행 이슈, 보고 사항 정리

The screenshot shows a Notion page titled "SSAFY 공통 PJT". At the top is a large, scenic image of a lake and mountains at night. Below the title, there are two introductory cards:

- 👉 든든한 팀장 "서요셉" 힙합킹 "안대현" 비전공실력킹 "곽호근" 조율정리킹 "배수연" 긍정킹 "서민영" 마지막한줄기빛 "윤영은"
- 👉 1등 예약 PJT 팀 Notion입니다.

The main content area is divided into several sections:

- 회의록**: A weekly calendar view for July 2021. The days of the week are labeled in Korean: 일 (Monday), 월 (Tuesday), 화 (Wednesday), 수 (Thursday), 목 (Friday), 금 (Saturday), and 토 (Sunday). The dates range from 27 to 31 July, with 28, 29, 30, and 31 being Saturday. The 27th and 28th are empty. The 29th has a note: "첫 회의!". The 29th, 30th, and 31st each have multiple overlapping boxes representing different meetings and tasks. The 29th box includes "1회 컨설턴트 미팅", "2회 컨설턴트 미팅", and "3회 컨설턴트 미팅". The 30th box includes "4회 컨설턴트 미팅". The 31st box includes "6회 컨설턴트 미팅", "7회 컨설턴트 미팅", and "8회 컨설턴트 미팅".
- 중요사항**: A sidebar with two items:
 - 팀 내 정책 (Team Internal Policies)
 - 코딩 컨벤션 (Coding Conventions)
- 프로젝트 구조도**: A sidebar with two items:
 - FE (Frontend)
 - BE (Backend)
- 설계**: A sidebar with five items:
 - UI / UX 와이어프레임 (UI / UX Wireframe)
 - DB ERD (Database Entity Relationship Diagram)
 - API 목록 (API List)
 - AWS (Amazon Web Services)
- 참고**: A sidebar with one item: "공유정보" (Shared Information).

04

Part별 진행 사항

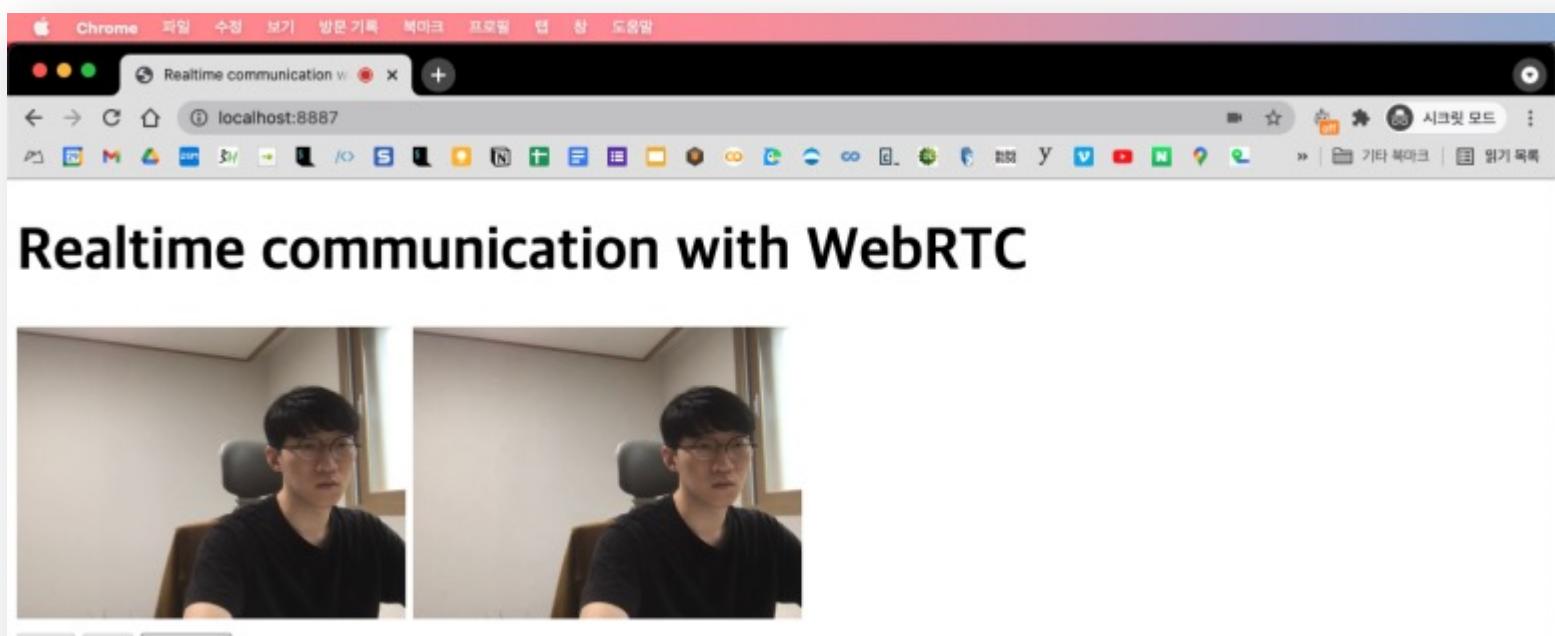
WebRTC
Backend
Frontend

04 Part별 진행 사항

WebRTC - Codelab

Kurento, Openvidu와 같은 Framework를 사용하기 전에,

WebRTC에 대해 더 이해하고자 Codelab 공식 문서 학습과 Tutorial 실습을 진행



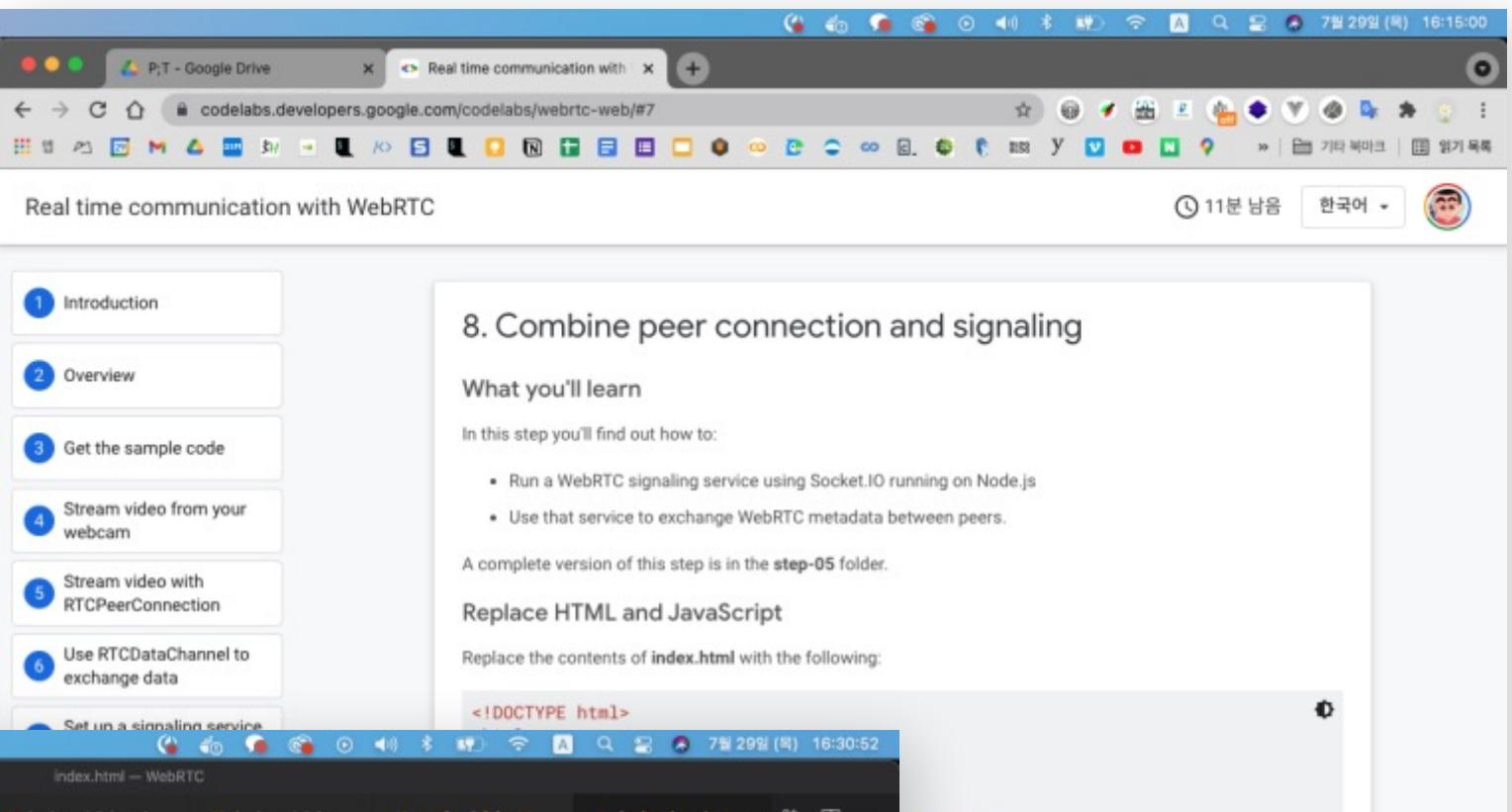
The screenshot shows a browser window titled "Realtime communication with WebRTC" displaying a video call interface. Two video feeds are visible: one from the user's own camera and one from a remote participant. Below the video feeds are three buttons: "Start", "Call", and "Hang Up". The "Call" button is currently highlighted.

The browser's address bar shows the URL "localhost:8887".

The right side of the image shows a code editor interface for a WebRTC project. The Explorer sidebar lists files such as index.js, package.json, step-06, work, css, js, lib, main-01.js, main-03.js, main-04.js, main-05.js, main-06.js, main.js, node_modules, index-03.html, index-04.html, index-05.html, index-06.html, index-07.html, index-08.html, index-09.html, index-10.html, index-11.html, index-12.html, index-13.html, index-14.html, index-15.html, index-16.html, index-17.html, index-18.html, index-19.html, index-20.html, index-21.html, index-22.html, index-23.html, index-24.html, index-25.html, index-26.html, index-27.html, index-28.html, index-29.html, index-30.html, index-31.html, index-32.html, index-33.html, index-34.html, index-35.html, index-36.html, index-37.html, index-38.html, package-lock.json, and server.js. The main editor area displays the content of index.html, which includes HTML, CSS, and JavaScript code for setting up a WebRTC peer connection.

The terminal at the bottom shows the command line output:

```
yeonglong@yeonglong-ui-MacBook-Pro WebRTC % cd WebRTC\ codelab\ examples  
yeonglong@yeonglong-ui-MacBook-Pro WebRTC codelab examples % cd work  
yeonglong@yeonglong-ui-MacBook-Pro work % node server.js  
[*] Listening on 0.0.0.0:8887
```



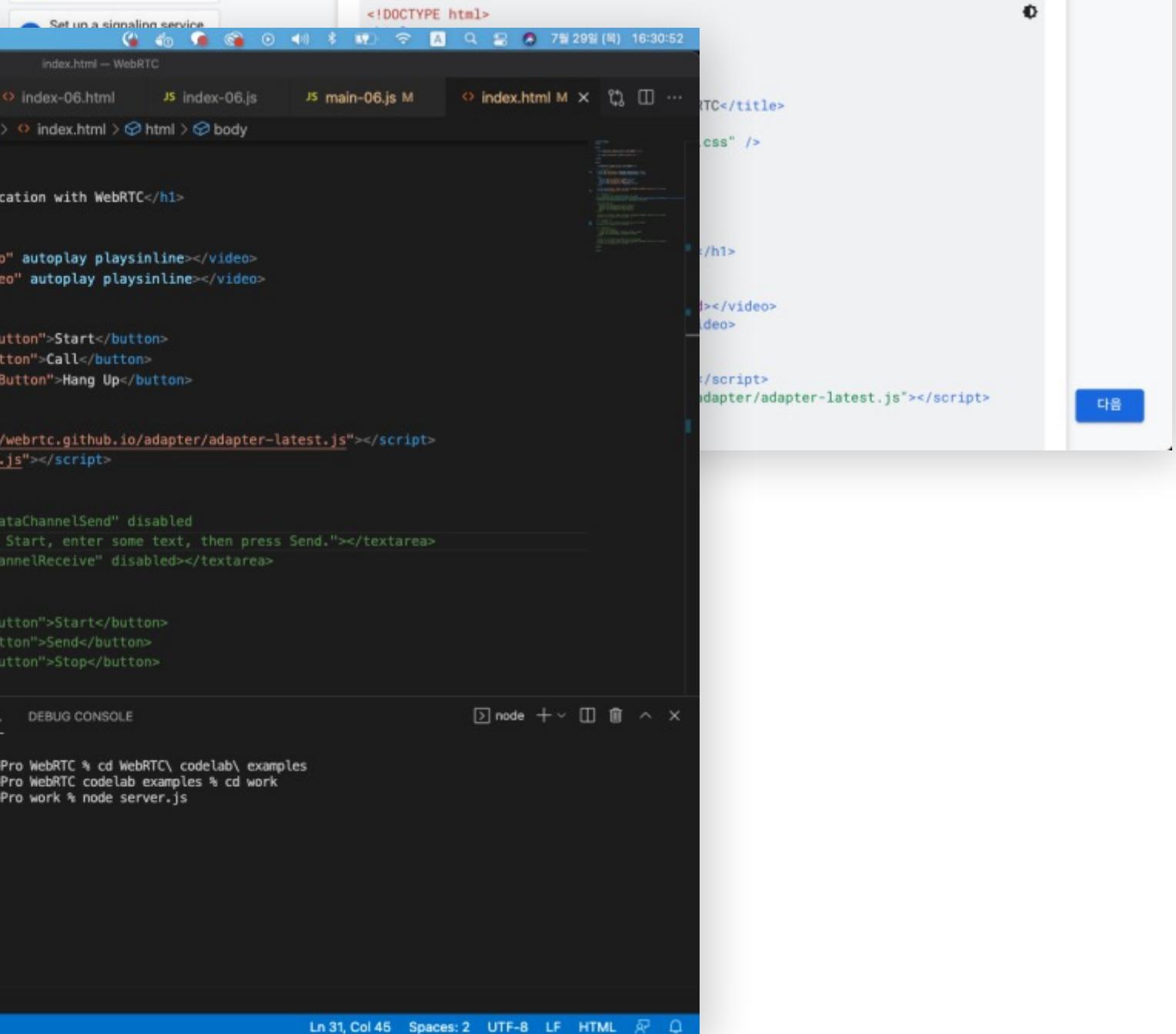
The screenshot shows a web browser window titled "Real time communication with WebRTC". On the left, a vertical navigation menu lists steps 1 through 6, each with a blue circular icon. Step 7 is partially visible. Step 8 is currently selected and expanded, titled "8. Combine peer connection and signaling".

The "What you'll learn" section states: "In this step you'll find out how to:"

- Run a WebRTC signaling service using Socket.IO running on Node.js
- Use that service to exchange WebRTC metadata between peers.

The "Replace HTML and JavaScript" section instructs: "Replace the contents of index.html with the following:"

```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <title>Realtime communication with WebRTC</title>  
    <link href="css/index.css" type="text/css" rel="stylesheet" />  
  </head>  
  <body>  
    <h1>Realtime communication with WebRTC</h1>  
    <!-- step 02 -->  
    <video id="localVideo" autoplay playsinline></video>  
    <video id="remoteVideo" autoplay playsinline></video>  
    <br>  
    <div>  
      <button id="startButton">Start</button>  
      <button id="callButton">Call</button>  
      <button id="hangupButton">Hang Up</button>  
    </div>  
    <script src="https://webrtc.github.io/adapter/adapter-latest.js"></script>  
    <script src="js/main.js"></script>  
    <br>  
    <!-- step 03 -->  
    <!-- <textarea id="dataChannelSend" disabled placeholder="Press Start, enter some text, then press Send."></textarea> -->  
    <!-- <textarea id="dataChannelReceive" disabled></textarea> -->  
    <br>  
    <div id="buttons">  
      <button id="startButton">Start</button>  
      <button id="sendButton">Send</button>  
      <button id="closeButton">Stop</button>  
    </div>  
  </body>  
</html>
```

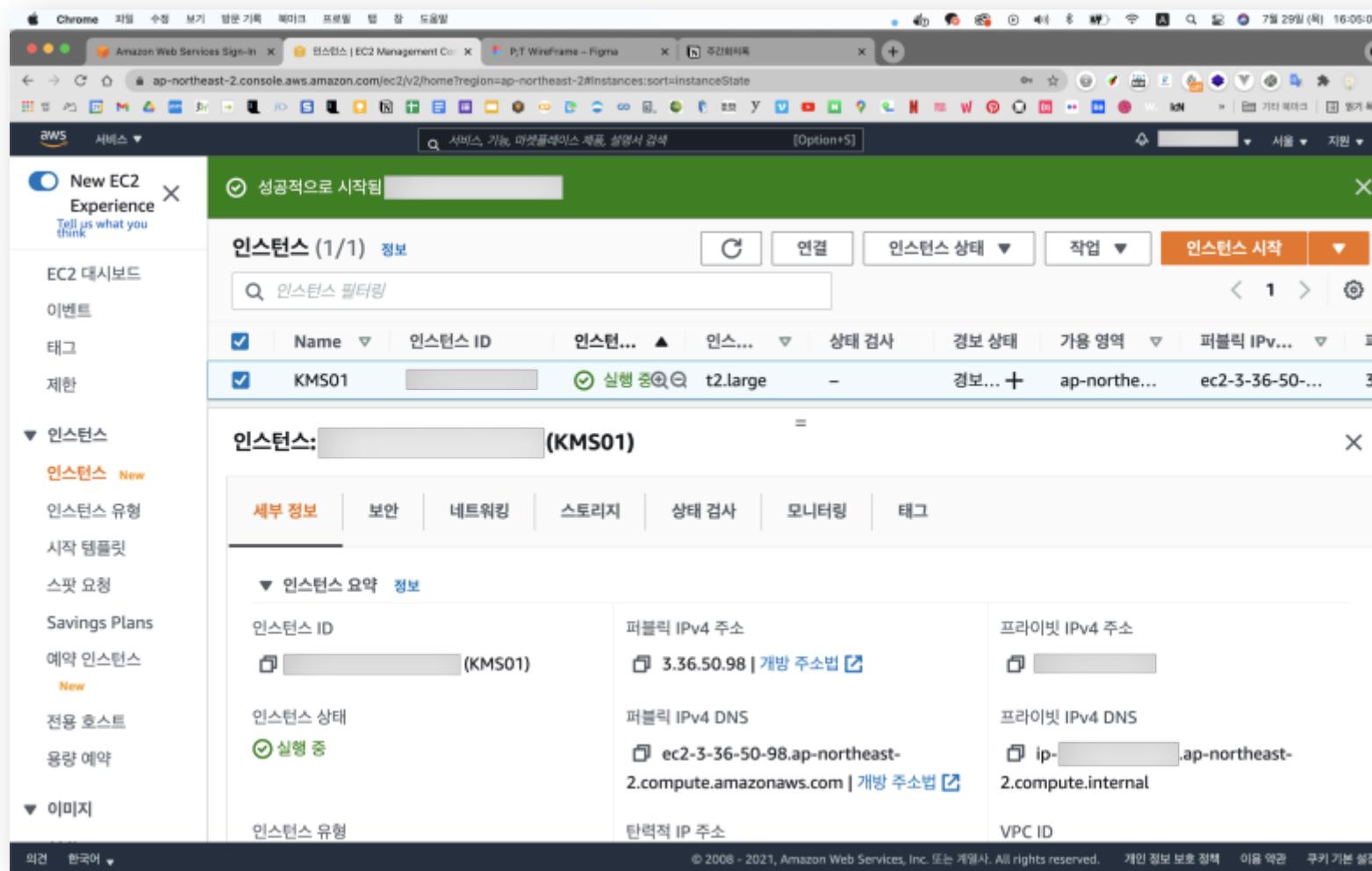


The screenshot shows a code editor with the file "index.html" open. The code is identical to the one shown in the browser screenshot above, demonstrating the implementation of a WebRTC peer connection and signaling.

04 Part별 진행 사항

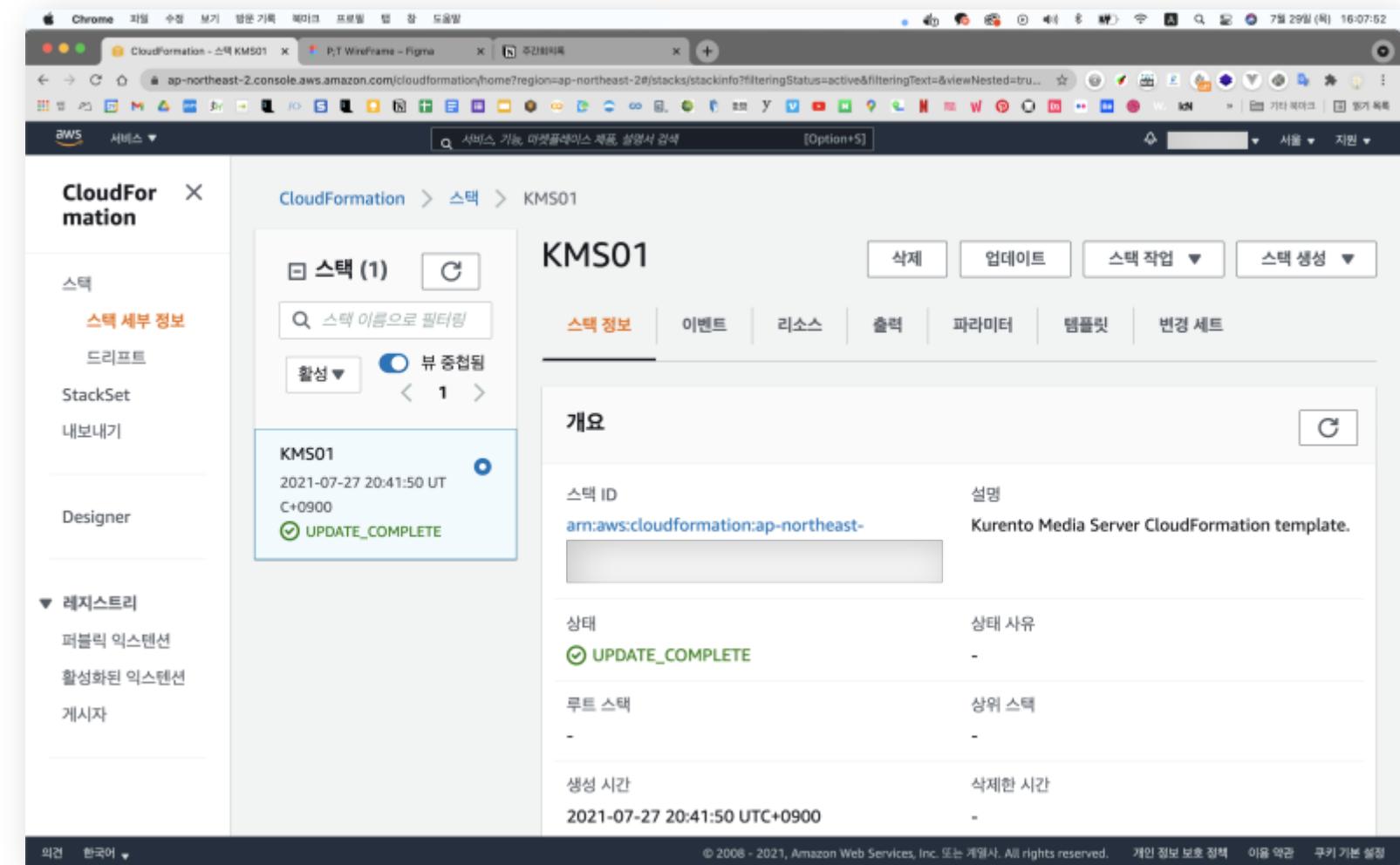
WebRTC - Kurento 사용을 위한 환경설정

1. 개인 AWS Instance & Kurento Stack 생성



The screenshot shows the AWS EC2 Management Console. A modal window titled "성공적으로 시작됨" (Successfully started) is displayed, indicating the creation of a new EC2 instance. The instance details are as follows:

- Instance ID:** KMS01
- Instance Type:** t2.large
- Region:** ap-northeast-2
- Public IP:** ec2-3-36-50-2.compute.amazonaws.com
- Public DNS:** ip-36-50-98.ap-northeast-2.compute.internal
- VPC ID:** 3.36.50.98 | 개방 주소법



The screenshot shows the AWS CloudFormation console. The KMS01 stack status is listed as "UPDATE_COMPLETE". The stack was created on 2021-07-27 at 20:41:50 UTC+0900. The stack ID is arn:aws:cloudformation:ap-northeast-2:0900. The stack description is "Kurento Media Server CloudFormation template".

* SSAFY 측에서 제공한 서버를 사용하기 전에, 개인 Instance와 Kurento Stack을 생성하여 Test를 진행

04 Part별 진행 사항

WebRTC - Kurento 사용을 위한 환경설정

2. ssh 접속

```
yeonglong@yeonglong-ui-MacBook-Pro ~ % ssh -i Documents/kms02.pem ubuntu@52.78.25.120
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1087-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: a minute https://ubuntu.com/advantage

166 packages can be updated. 인스턴스 상태 ▾
123 updates are security updates.

*** System restart required ***      퍼블릭 IPv4 주소
                                     프라이빗 IPv4 주소
                                     □ 172.31
                                     The programs included with the Ubuntu system are free software; 주소
                                     the exact distribution terms for each program are described in the
                                     individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.                      퍼블릭 IPv4 DNS
                                     프라이빗 IPv4 DNS
                                     □ ip-172
                                     To run a command as administrator (user "root"), use "sudo <command>".
                                     See "man sudo_root" for details.          northeast-
                                     northeast-
                                     2.compute.amazonaws.com |
```

3. Docker 설치

```
[1]
Get:4 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 runc amd64 1.0.0~rc7+git20190403.029124da-0ubuntu1~16.04.4 [1,890 kB]
Get:5 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 containerd amd64 1.2.6-0ubuntu1~16.04.6 [19.9 MB]
Get:6 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 docker.io amd64 18.09.7-0ubuntu1~16.04.7 [30.5 MB]
Get:7 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 ubuntu-fan all 0.12.8~16.04.3 [35.1 kB]
Fetched 52.4 MB in 2s (21.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 74480 files and directories currently installed.)
Preparing to unpack .../pigz_2.3.1-2_amd64.deb ...
Unpacking pigz (2.3.1-2) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../bridge-utils_1.5-9ubuntu1_amd64.deb ...

Adding group `docker' (GID 120) ...
Done.
Setting up ubuntu-fan (0.12.8~16.04.3) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.21) ...
ubuntu@ip-172-31-3-212:~$ docker --version
Docker version 18.09.7, build 2d0083d
ubuntu@ip-172-31-3-212:~$
```

04 Part별 진행 사항

WebRTC - Kurento 사용을 위한 환경설정

4. Kurento Container 설치

```
ubuntu@ip-172-31-3-212:~$ docker pull kurento/kurento-media-server:latest
latest: Pulling from kurento/kurento-media-server.amazonaws.com |
4007a89234b4: Pull complete
5dfa26c6b9c9: Pull complete
0ba7bf18aa40: Pull complete
4c6ec688ebe3: Pull complete
4a42be364f88: Pull complete
2d4af0a8e4fc: Pull complete
f1a3089b94df: Pull complete
a1ca7ec2b6bc: Pull complete
861752364513: Pull complete
1aba1253d4a6: Pull complete
0a31d5ca11ee: Pull complete
e52492c47085: Pull complete
5cf9c63e08f5: Pull complete
1834c5f6eb61: Pull complete
Digest: sha256:aff1bacd760253dc573935e92bb941b04733ac75e5af726c3634814329dfc026
Status: Downloaded newer image for kurento/kurento-media-server:latest
```

5. Coturn 설치

```
ubuntu@ip-172-31-3-212:~$ sudo apt-get update
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:5 http://ubuntu.openvidu.io/6.16.0 xenial InRelease
Fetched 216 kB in 0s (296 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-3-212:~$ sudo apt-get install --no-install-recommends --yes \
> coturn
coturn を 위해 AWS
Reading package lists... Done
Building dependency tree
Reading state information... Done
coturn is already the newest version (4.5.0.3-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 159 not upgraded.
```

04 Part별 진행 사항

WebRTC - Kurento 사용을 위한 환경설정

6. STUN / TURN 설정

```
;; STUN server IP address.
;; The ICE process uses STUN to punch holes through NAT firewalls.
;;
;; You don't need to configure both STUN and TURN, because TURN already includes
;; STUN functionality.
;;
;; <stunServerAddress> MUST be an IP address; domain names are NOT supported.
;;
;; You need to use a well-working STUN server. Use this to check if it works:
;; https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/
;; From that check, you should get at least one Server-Reflexive Candidate
;; (type "srflx").
;;
;; stunServerAddress=52.78.25.120
;; stunServerPort=3478
;;
;; TURN server URL:92c739a0c5
;; When STUN is not enough to open connections through some NAT firewalls,
;; using TURN is the remaining alternative.
;;
;; You don't need to configure both STUN and TURN, because TURN already includes
;; STUN functionality.
;;
;; The provided URL should follow one of these formats:
;; * user:password@ipaddress:port    2.compute.amazonaws.com/
;; * user:password@ipaddress:port?transport=[udp|tcp|tls]
;;
;; <ipaddress> MUST be an IP address; domain names are NOT supported.
;; <transport> is OPTIONAL. Possible values: udp, tcp, tls. Default: udp.
;;
;; You need to use a well-working TURN server. Use this to check if it works: vpc-c552c3ae
;; https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/
;;
;; From that check, you should get at least one Server-Reflexive Candidate
;; (type "srflx") AND one Relay Candidate (type "relay").
;;
;; AWS
turnURL=kurento:kurento@52.78.25.120:3478?transport=udp
```

7. Git / Maven / JDK 설치

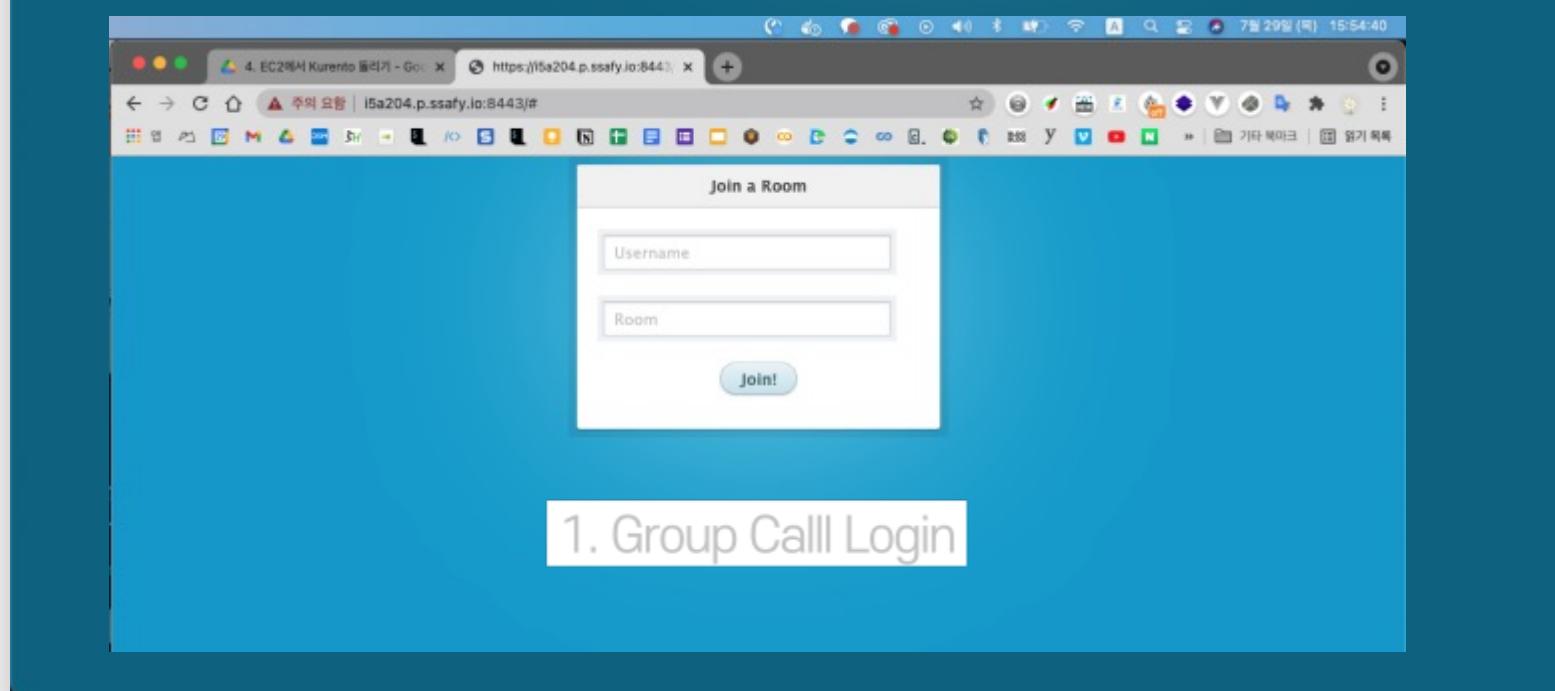
```
ubuntu@ip-172-31-3-212:~$ sudo apt-get update && sudo apt-get install --no-install-recommends --yes \
> git \
> default-jdk \
> maven
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ubuntu.openvidu.io/6.16.0 xenial InRelease
Fetched 325 kB in 1s (245 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
default-jdk is already the newest version (2:1.8-56ubuntu2).
maven is already the newest version (3.3.9-3).
git is already the newest version (1:2.7.4-0ubuntu1.10).
```

04 Part별 진행 사항

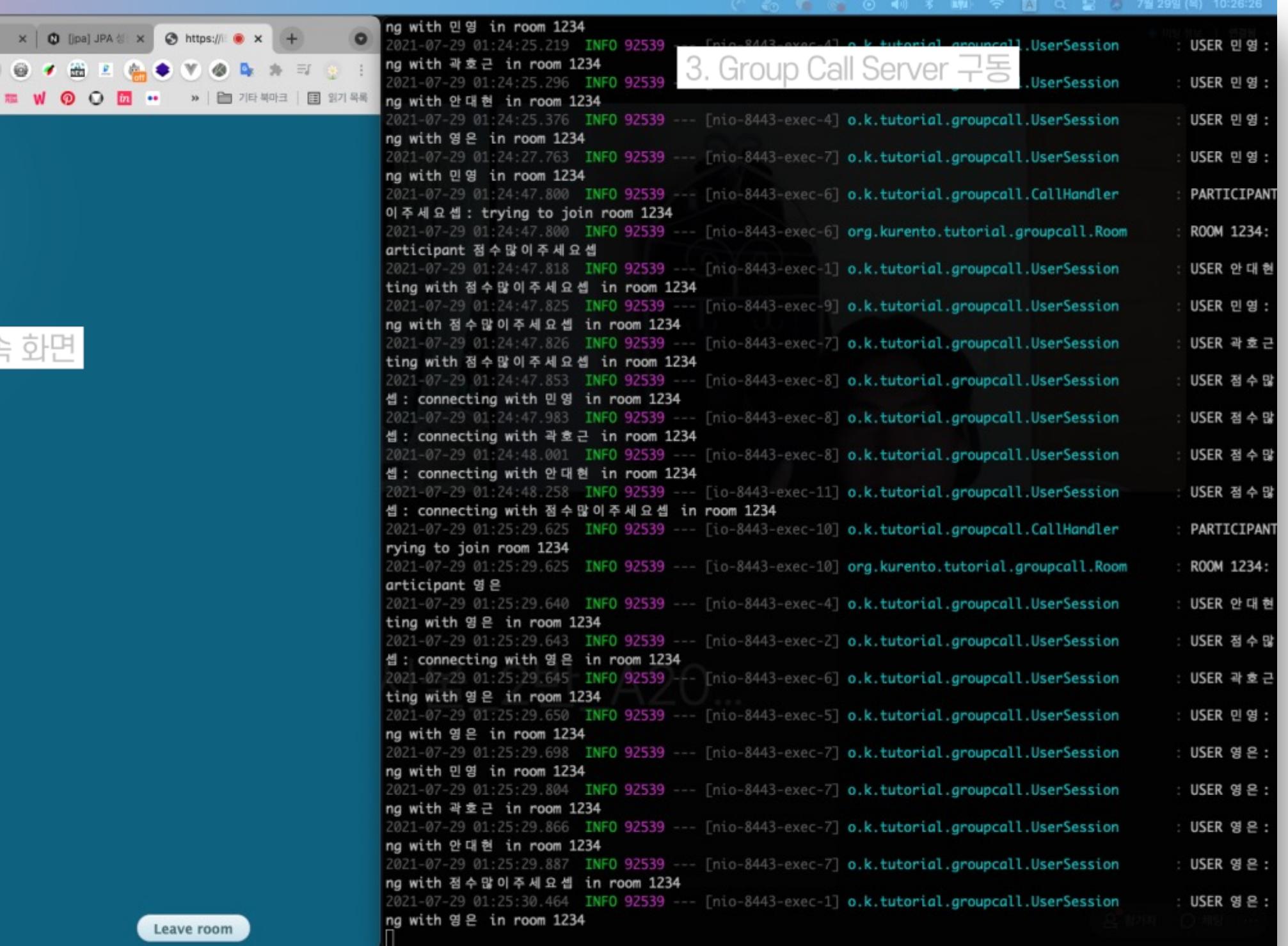
WebRTC - EC2에서 Kurento 실행



2. Group Call 접속 화면



1. Group Call Login



3. Group Call Server 구동

```
ng with 민영 in room 1234
2021-07-29 01:24:25.219 INFO 92539 --- [nio-8443-exec-4] o.k.tutorial.groupcall.UserSession
ng with 곽호근 in room 1234
2021-07-29 01:24:25.296 INFO 92539 --- [nio-8443-exec-4] o.k.tutorial.groupcall.UserSession
ng with 안대현 in room 1234
2021-07-29 01:24:25.376 INFO 92539 --- [nio-8443-exec-4] o.k.tutorial.groupcall.UserSession
ng with 영은 in room 1234
2021-07-29 01:24:27.763 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ng with 민영 in room 1234
2021-07-29 01:24:47.800 INFO 92539 --- [nio-8443-exec-6] o.k.tutorial.groupcall.CallHandler
이주세요셉 : trying to join room 1234
2021-07-29 01:24:47.800 INFO 92539 --- [nio-8443-exec-6] org.kurento.tutorial.groupcall.Room
participant 접수 많이주세요셉
2021-07-29 01:24:47.818 INFO 92539 --- [nio-8443-exec-1] o.k.tutorial.groupcall.UserSession
ting with 접수 많이주세요셉 in room 1234
2021-07-29 01:24:47.825 INFO 92539 --- [nio-8443-exec-9] o.k.tutorial.groupcall.UserSession
ng with 접수 많이주세요셉 in room 1234
2021-07-29 01:24:47.826 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ting with 접수 많이주세요셉 in room 1234
2021-07-29 01:24:47.853 INFO 92539 --- [nio-8443-exec-8] o.k.tutorial.groupcall.UserSession
셉 : connecting with 민영 in room 1234
2021-07-29 01:24:47.983 INFO 92539 --- [nio-8443-exec-8] o.k.tutorial.groupcall.UserSession
셉 : connecting with 곽호근 in room 1234
2021-07-29 01:24:48.001 INFO 92539 --- [nio-8443-exec-8] o.k.tutorial.groupcall.UserSession
셉 : connecting with 안대현 in room 1234
2021-07-29 01:24:48.258 INFO 92539 --- [nio-8443-exec-11] o.k.tutorial.groupcall.UserSession
셉 : connecting with 접수 많이주세요셉 in room 1234
2021-07-29 01:25:29.625 INFO 92539 --- [nio-8443-exec-10] o.k.tutorial.groupcall.CallHandler
rying to join room 1234
2021-07-29 01:25:29.625 INFO 92539 --- [nio-8443-exec-10] org.kurento.tutorial.groupcall.Room
participant 영은
2021-07-29 01:25:29.640 INFO 92539 --- [nio-8443-exec-4] o.k.tutorial.groupcall.UserSession
ting with 영은 in room 1234
2021-07-29 01:25:29.643 INFO 92539 --- [nio-8443-exec-2] o.k.tutorial.groupcall.UserSession
셉 : connecting with 영은 in room 1234
2021-07-29 01:25:29.645 INFO 92539 --- [nio-8443-exec-6] o.k.tutorial.groupcall.UserSession
ting with 영은 in room 1234
2021-07-29 01:25:29.650 INFO 92539 --- [nio-8443-exec-5] o.k.tutorial.groupcall.UserSession
ng with 영은 in room 1234
2021-07-29 01:25:29.698 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ng with 민영 in room 1234
2021-07-29 01:25:29.804 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ng with 곽호근 in room 1234
2021-07-29 01:25:29.866 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ng with 안대현 in room 1234
2021-07-29 01:25:29.887 INFO 92539 --- [nio-8443-exec-7] o.k.tutorial.groupcall.UserSession
ng with 접수 많이주세요셉 in room 1234
2021-07-29 01:25:30.464 INFO 92539 --- [nio-8443-exec-1] o.k.tutorial.groupcall.UserSession
ng with 영은 in room 1234
```

04 Part별 진행 사항

Backend - Postman API Test

기능: 회원가입 - 닉네임, 이메일 중복확인

상태코드: 중복될 때(411), 중복되지 않을 때(200)

The screenshot shows the Postman application interface. A POST request is being made to `http://13.124.187.160:8080/v1/users/`. The request body is a JSON object containing user information. The response status is 200, and the message indicates a successful registration.

```
1 {  
2   "userGender": "011",  
3   "userName": "서요셉",  
4   "userEmail": "tjdytpq@gmail.com",  
5   "userPwd": "0310@tjdytpq",  
6   "userType": "001",  
7   "userProfile": "",  
8   "userNickname": "hoho",  
9   "userDesc": "",  
10  "userPhone": "01099534290"  
11 }
```

Body Cookies Headers (12) Test Results
Pretty Raw Preview Visualize JSON

```
1 {  
2   "message": "Email과 닉네임 모두 사용중입니다.",  
3   "statusCode": 411  
4 }
```

기능: 회원가입 - 닉네임 중복확인

상태코드: 중복될 때(410), 중복되지 않을 때(200)

The screenshot shows the Postman application interface. A POST request is being made to `http://13.124.187.160:8080/v1/users/`. The request body is identical to the one in the previous screenshot. The response status is 411, indicating that the nickname is already in use.

```
1 {  
2   "userGender": "011",  
3   "userName": "서요셉",  
4   "userEmail": "tjdytpq@gmail.com",  
5   "userPwd": "0310@tjdytpq",  
6   "userType": "001",  
7   "userProfile": "",  
8   "userNickname": "hoho",  
9   "userDesc": "",  
10  "userPhone": "01099534290"  
11 }
```

Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON

```
1 {  
2   "message": "다른 회원이 사용하고 계신 닉네임입니다.",  
3   "statusCode": 410  
4 }
```

04 Part별 진행 사항

Backend - Postman API Test

기능: 회원가입 - 이메일 중복확인

상태코드: 중복될 때(409), 중복되지 않을 때(200)

The screenshot shows the Postman application interface. A POST request is being made to `http://13.124.187.160:8080/v1/users/`. The request body is a JSON object containing user registration details. The response status is 409 Conflict, indicating that the email address is already registered.

```
1 {  
2   "userGender": "011",  
3   "userName": "서요셉",  
4   "userEmail": "tjdytpq@gmail.com",  
5   "userPwd": "0310@tjdytpq",  
6   "userType": "001",  
7   "userProfile": "",  
8   "userNickname": "hohoho",  
9   "userDesc": "",  
10  "userPhone": "01099534290"  
11 }
```

Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON

```
1 {  
2   "message": "이미 가입된 Email입니다.",  
3   "statusCode": 409  
4 }
```

기능: 회원가입 - 회원 등록

상태코드: 실패(409), 성공(200)

The screenshot shows the Postman application interface. A POST request is being made to `http://13.124.187.160:8080/v1/users/`. The request body is a JSON object containing user registration details. The response status is 200 OK, indicating that the user was successfully registered.

```
1 {  
2   "userGender": "011",  
3   "userName": "서요셉",  
4   "userEmail": "tjdytpq@gmail.com",  
5   "userPwd": "0310@tjdytpq",  
6   "userType": "001",  
7   "userProfile": "",  
8   "userNickname": "hoh",  
9   "userDesc": "",  
10  "userPhone": "01099534290"  
11 }
```

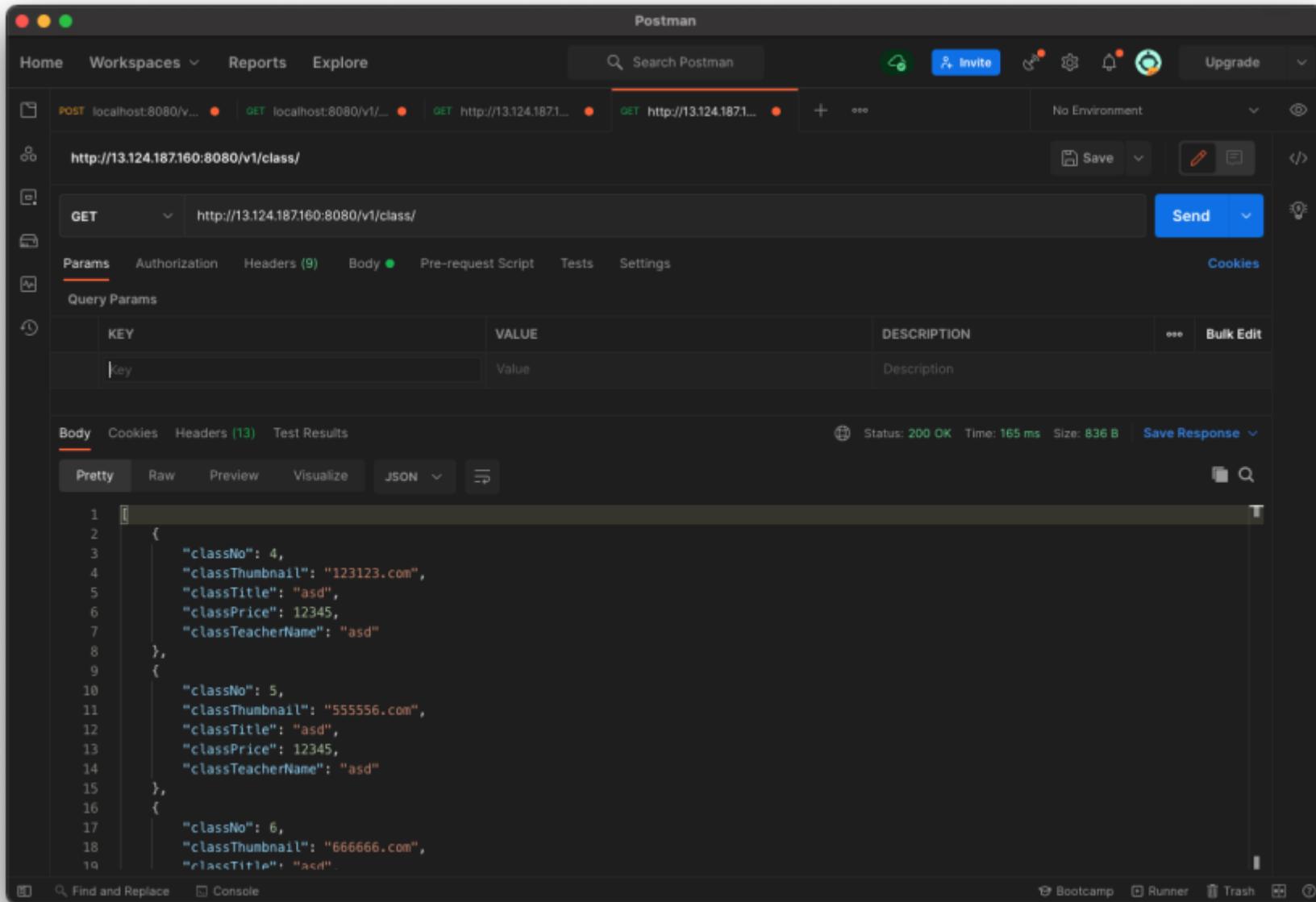
Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON

```
1 {  
2   "message": "환영합니다. 회원가입에 성공하셨습니다.",  
3   "statusCode": 200  
4 }
```

04 Part별 진행 사항

Backend - Postman API Test

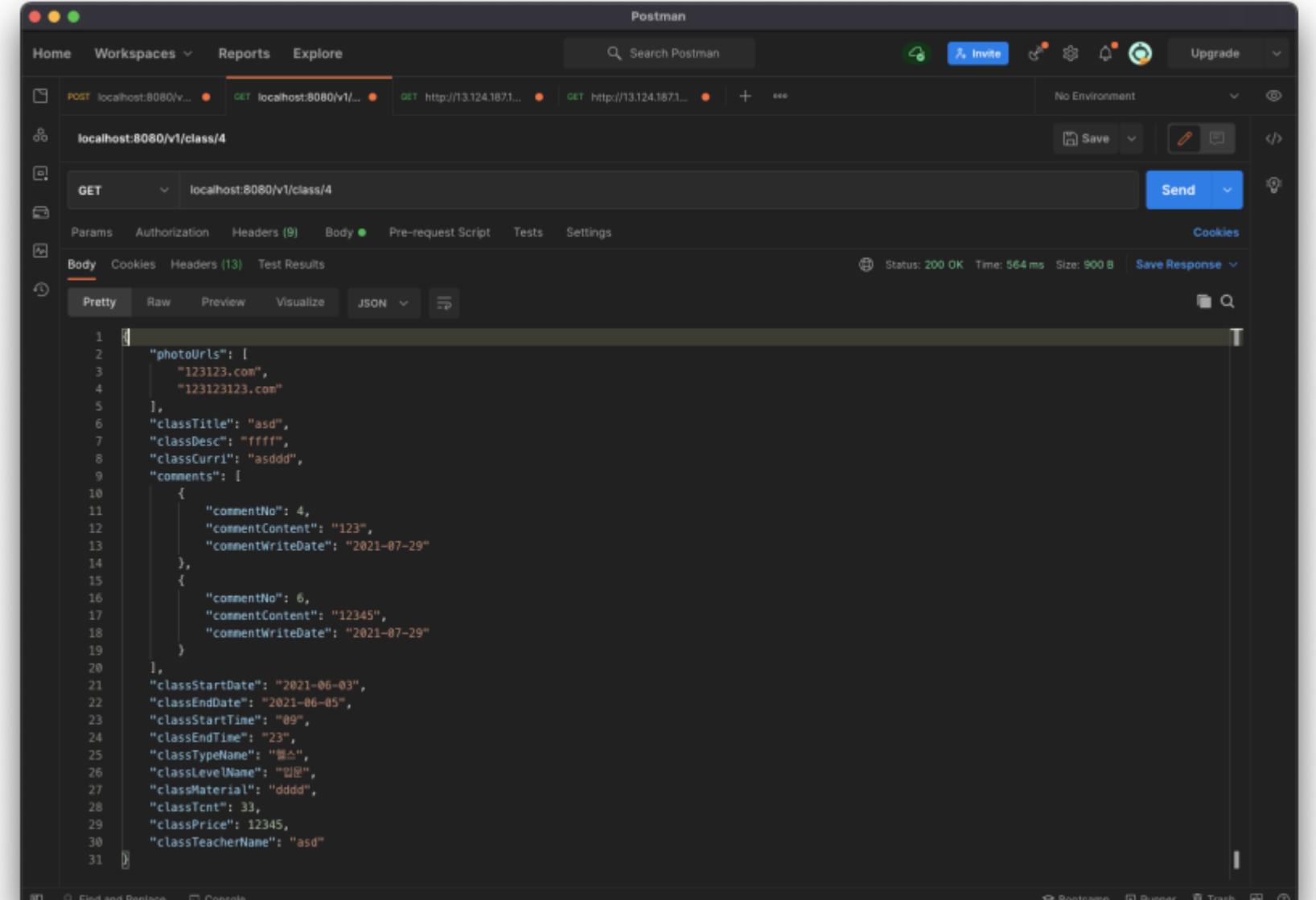
기능: 클래스 - 전체 클래스 목록 가져오기
상태코드: 성공(200)



The screenshot shows the Postman application interface. A GET request is made to `http://13.124.187.160:8080/v1/class/`. The response body is displayed in JSON format, showing a list of three class objects. Each object contains fields like classNo, classThumbnail, classTitle, classPrice, and classTeacherName.

```
1 [ {  
2   "classNo": 4,  
3   "classThumbnail": "123123.com",  
4   "classTitle": "asd",  
5   "classPrice": 12345,  
6   "classTeacherName": "asd"  
7 },  
8 {  
9   "classNo": 5,  
10  "classThumbnail": "555556.com",  
11  "classTitle": "asd",  
12  "classPrice": 12345,  
13  "classTeacherName": "asd"  
14 },  
15 {  
16   "classNo": 6,  
17   "classThumbnail": "666666.com",  
18   "classTitle": "asd"  
19 } ]
```

기능: 클래스 - 특정 클래스 상세 정보 가져오기
상태코드: 성공(200)



The screenshot shows the Postman application interface. A GET request is made to `http://13.124.187.160:8080/v1/class/4`. The response body is displayed in JSON format, showing a single class object with additional details such as photoUrls, classTitle, classDesc, classCur1, and comments.

```
1 {  
2   "photoUrls": [  
3     "123123.com",  
4     "123123123.com"  
5   ],  
6   "classTitle": "asd",  
7   "classDesc": "ffff",  
8   "classCur1": "asddd",  
9   "comments": [  
10    {  
11      "commentNo": 4,  
12      "commentContent": "123",  
13      "commentWriteDate": "2021-07-29"  
14    },  
15    {  
16      "commentNo": 6,  
17      "commentContent": "12345",  
18      "commentWriteDate": "2021-07-29"  
19    }  
20  ],  
21  "classStartDate": "2021-06-03",  
22  "classEndDate": "2021-06-05",  
23  "classStartTime": "09",  
24  "classEndTime": "23",  
25  "classTypeName": "苒々",  
26  "classLevelName": "입문",  
27  "classMaterial": "dddd",  
28  "classTcnt": 33,  
29  "classPrice": 12345,  
30  "classTeacherName": "asd"  
31 }
```

04 Part별 진행 사항

Frontend - Home

Top 10 인기 클래스

이벤트 클래스

The screenshot shows a fitness application interface. At the top, there is a navigation bar with a logo (P:T), search fields, and user account options. Below the header, a large banner image shows a person exercising on a stationary bike. Underneath the banner, the text "Top10 인기 클래스" (Top 10 Popular Classes) is displayed, followed by four cards representing different fitness classes:

Image	Name	Price	Action
	홈트계의 끝판왕 '힙으뜸'	월 46,800원 (5개월)	선택하기
	SNPE 바디리셋 프로젝트. 척추운동으…	월 46,800원 (5개월)	선택하기
	천재 사업가 '대학생김머신'의 무엇이든…	월 46,800원 (5개월)	선택하기
	이성과 감성을 넘나들다. 시선을 사로잡…	월 46,800원 (5개월)	선택하기

04 Part별 진행 사항

Frontend - 클래스 상세

해당 클래스의 이미지, 클래스 설명, 커뮤니티(후기) 등의
상세 정보 확인 가능

PIT

Select 검색 상세 검색 조건

장바구니 마이클래스

Home 1:1 1:N 이벤트



클래스 설명 커리큘럼 커뮤니티

클래스 설명

달심쌤과 함께 퇴근 후 요가파이야

퇴근 이후에 코어 근육 강화를 위한 요가 수업입니다. 늦은 저녁 시간에 강의가 진행됩니다.

커리큘럼

1주차 : 수강생의 상태 파악
2주차 : --
3주차 : --
4주차 : --
5주차 : --
6주차 : --
7주차 : --
8주차 : --
9주차 : --
10주차 : --
11주차 : --
12주차 : --

커뮤니티

id123 2021-07-29

정말 좋아요! 시간도 정말 적절하고 강사님께서 배려심 넘치세요!! 맨날 앉아있기만 해서 허리가 정말 안 좋았는데 달심쌤 수업 듣고 자세도 훨씬 좋아졌어요! 강력추천 드립니다!!

달심쌤과 함께 퇴근 후 요가파이야

구매정보

8월 8일부터 수강 가능

□ 대분류 요가
△ 준비물 요가매트, 요가복
☆ 레벨 초보
△ 기간 3달
☆ 수업 수 24

▣ 금액 총 300000원

월 100000원

신청하기

04 Part별 진행 사항

Frontend - 마이클래스

현재 수강중인 클래스 목록
확인과 화상PT룸 입장 가능

P:T

Select ▾ 검색 상세 검색 조건

장바구니 마이클래스

Home 1:1 ▾ 1:N ▾ 이벤트

□ 내 클래스

수강중 클래스

수강완료 클래스

녹화된 수업

피드백

★ 찜한 클래스

○ 내 정보

수강중 클래스

요가 강사: 심으뜸
홈트계의 끝판왕 '힙으뜸'
검증된 운동 유튜버의 업데이트 된 커리큘럼!

입장하기

< June 2021 >

SU	MO	TU	WE	TH	FR	SA
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

< June 2021 >

SU	MO	TU	WE	TH	FR	SA
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Copyright © A204 All Rights Reserved.

HG SY MY YS DH YE Presents.

감사합니다

서울 2반 4조 | 서요셉 곽호근 서민영 안대현 윤영은