LivekitDev 프로젝트 분석

| 작성일: 2026-07-19 · 한글 복구: 2026-09-07
| 작업 경로: C:₩Developments₩LivekitDev
| 아래는 초기 데모 분석 기록입니다. 문서 저장 위치와 이후 배포 변경은 별도로 표시했습니다.

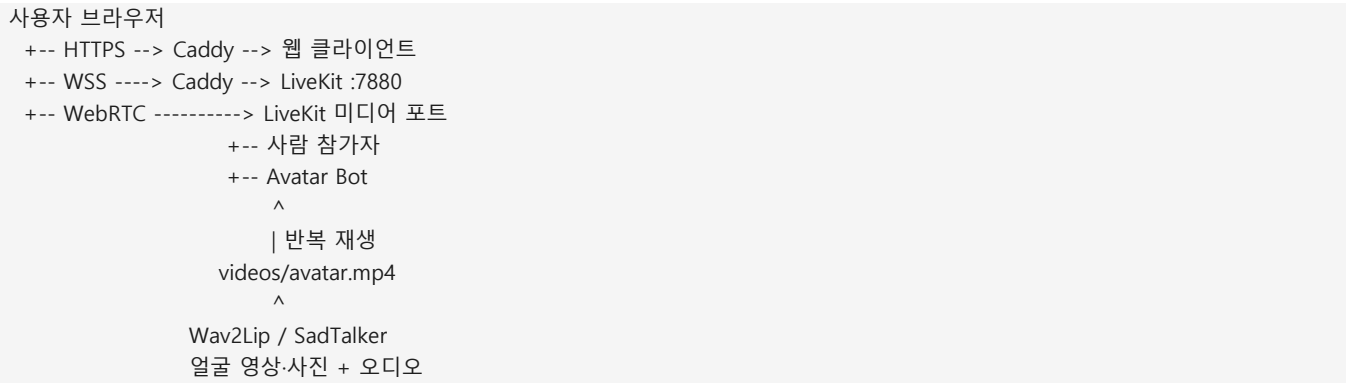
1. 한 줄 요약

실시간 화상통화(LiveKit)와 오프라인 AI 토킹 아바타를 묶은 데모입니다. Wav2Lip 또는 SadTalker로 영상을 만든 뒤 Avatar Bot이 LiveKit 방에서 상주 참가자로 송출하고, Flutter 웹 클라이언트가 같은 방에 접속합니다.

2. 전체 구조

경로	역할	성격
livekit/	LiveKit SFU 서버, Go, v1.13.4	실시간 미디어 중계
livekit_video_call/	Flutter 웹 화상 클라이언트	사용자 UI
avatar_bot/	아바타 영상 상주 송출	Python 참가자
Wav2Lip/	얼굴 영상과 오디오로 립싱크 생성	오프라인 생성
SadTalker/	사진과 오디오로 토킹헤드 생성	오프라인 생성
BasicSR/	화질 및 복원 프레임워크	라이브러리
videos/	원본 얼굴 영상 및 오디오	입력 자료
Caddyfile, caddy.exe	HTTPS 리버스 프록시	외부 접속
start_all.bat	기존 일괄 실행기	레거시 기동
docs-site/public/downloads/	통합 Markdown 및 PDF	문서 원본
docs-site/src/content/docs/	Astro Starlight 페이지	웹 문서

3. 아키텍처



핵심은 미리 만든 토킹 영상을 실시간 방에 계속 발행하는 방식입니다. 파일 수정 시간이 바뀌면 봇이 영상을 다시 읽습니다.

초기 Caddy 기록에는 /videocall/의 Flutter 정적 파일, /rtc와 7443의 LiveKit 프록시, Reflex용 3000/8000 경로가 있었습니다. 이후 구성은 3000 포트의 Meet를 사용합니다. 현재 배포는 Windows 서비스 문서를 참고합니다.

4. LiveKit 서버

- 공식 서버 소스와 Windows 실행 파일 livekit/bin/livekit-server.exe
- 기록된 업데이트: 2026-07-19, 1.9.11에서 1.13.4로 변경
- 기본 HTTP 및 WebSocket 포트: 7880
- 개발용 키는 Flutter 및 봇 설정과 일치해야 함
- Redis 및 클러스터 구성은 초기 최소 설정에 포함되지 않음

```
logging:
  level: info
keys:
  devkey: secret
```

5. Flutter 클라이언트

파일	역할
lib/main.dart	Material 3 테마 및 앱 시작
lib/pages/connect_page.dart	서버 URL, 방 이름, 닉네임 입력
lib/pages/room_page.dart	참가자 그리드, 미디어 및 아바타 제어
lib/utils/token_generator.dart	개발용 클라이언트 JWT 발급
lib/widgets/controls.dart	마이크, 카메라, 아바타 토글 및 퇴장
lib/widgets/participant_tile.dart	비디오 타일 및 발화 상태 표시

기본 접속 값과 옵션

- 서버: wss://meet.dev.tictechtoeai.com:7443
- 방: my-room
- 기록된 의존성: livekit_client ^2.6.3, dart_jsonwebtoken ^3.3.1
- 기록된 옵션: adaptiveStream, dynacast, 720p, simulcast, DTX

클라이언트에서 개발 키로 JWT에 서명하므로 운영용 구성에서는 서버 측 토큰 발급이 필요합니다.

아바타 DataChannel 프로토콜

Topic	방향	메시지
avatar-control	Flutter에서 Bot으로	video:on, video:off, audio:on, audio:off, status
avatar-status	Bot에서 Flutter로	비디오 및 오디오 상태 알림

Flutter는 입장 후 상태를 요청해 봇 상태와 UI를 맞춥니다.

6. Avatar Bot

- identity: avatar-bot, 표시명: Avatar
- avatar_bot/videos/avatar.mp4를 반복 송출
- PyAV 디코딩: 비디오 RGBA 및 오디오 s16 mono 48 kHz
- 파일 수정 시간 변경을 감지해 다시 불러오기
- 초기 분석의 비디오 비트레이트 상한: 1.5 Mbps
- 연결 종료 시 재접속 대기: 5초

• generate_test_video.py: OpenCV로 테스트 영상 생성

환경 변수	기본값
LIVEKIT_URL	ws://localhost:7880
LIVEKIT_API_KEY	devkey
LIVEKIT_API_SECRET	secret
ROOM_NAME	my-room

기록된 의존성은 livekit, livekit-api, opencv-python-headless, numpy, python-dotenv이며 코드에서 PyAV의 av 패키지도 사용합니다. 기존 requirements 프로젝트의 관리 방식 변경은 별도 확인 후 진행합니다.

7. 인프라

Caddy

초기 프록시 기록:

경로 또는 포트	대상
/rtc, /twirp/	LiveKit 7880
7443	LiveKit WSS
/videocall/*	초기 Flutter 정적 파일
/api/, /_event	초기 Reflex용 8000 예약 경로
기본 /	초기 3000 UI, 이후 Meet 사용

start_all.bat

1. WSL Ubuntu에서 Caddyfile을 복사하고 Caddy를 다시 불러옵니다.
2. LiveKit 서버를 시작합니다.
3. Avatar Bot 실행 부분은 REM으로 비활성화되어 있습니다.
4. 초기 안내 URL은 https://meet.dev.tictechtoeai.com/videocall/입니다.

이 실행기에는 D:₩LivekitDev 경로가 남아 있으므로 현재 C:₩Developments₩LivekitDev와 구분해야 합니다. 이후 서비스 구성에는 start_all_services.bat가 있습니다.

8. AI 생성 파이프라인

Wav2Lip

스크립트	기능
inference.py	기본 립싱크 생성
enhance_video.py	GFPGAN 얼굴 복원
enhance_codeformer.py	CodeFormer 복원
full_pipeline.py	립싱크 및 복원 과정 연결

기록된 체크포인트: Wav2Lip-SD-GAN.pt, Wav2Lip-SD-NOGAN.pt.
결과 파일 예시: resultrodu.mp4, whd_rodu_final.mp4.

입력 자료

파일	초기 분석에서의 용도
videos/avatar.mp4	메인 아바타 영상 후보, 당시 약 290 MB
videos/b.mp4, videos/w.mp4	짧은 얼굴 클립 테스트
videos/whd.mp4	고품질 얼굴 영상 입력
videos/rodu.wav	공통 구동 오디오

SadTalker

사진 한 장과 오디오로 3D 모션 토킹헤드를 생성합니다. 초기 기록에는 256/512 safetensors와 mapping 모델이 준비되어 있었으며 Wav2Lip보다 실행 결과 기록이 적었습니다.

BasicSR

초해상도와 복원 학습·추론 프레임워크이며 CodeFormer 등 복원 기능의 기반입니다.

9. 전체 작업 흐름

1. 얼굴 영상 또는 사진과 음성으로 오프라인 영상을 생성합니다.
2. 결과를 avatar_bot/videos/avatar.mp4에 배치합니다.
3. Caddy 및 LiveKit을 시작하고 봇을 실행합니다.
4. 사용자가 같은 방에 입장해 아바타를 보고 영상·음성을 제어합니다.

10. 기술 스택

영역	기술
미디어 SFU	LiveKit Server 1.13.4, Go
초기 클라이언트	Flutter Web 및 livekit_client
아바타 발행	Python, LiveKit SDK, PyAV, OpenCV
립싱크	Wav2Lip, PyTorch
얼굴 복원	GFPGAN, CodeFormer, BasicSR
대체 토킹헤드	SadTalker
프록시 및 TLS	Caddy
초기 인증	개발용 JWT
문서	Markdown, PDF, Astro Starlight

11. 초기 점검 당시 상태

구성 요소	기록된 상태
LiveKit 실행 파일 및 최소 설정	준비됨
Flutter UI 및 아바타 제어	구현 및 빌드 산출물 존재
Avatar Bot	구현됨
Wav2Lip 실험	여러 결과 영상 생성
SadTalker 체크포인트	다운로드됨

구성 요소	기록된 상태
Flutter 서버 측 토큰 발급	초기 구성에 없음
Reflex	초기 Caddy 경로만 존재, 해당 앱 소스 없음
Avatar Bot 자동 기동	기존 배치에서 주석 처리
경로 일관성	D 드라이브와 현재 C 드라이브 경로 혼재

이 표는 현재 프로세스 상태를 실시간 점검한 결과가 아닙니다.

12. 설계 의도

Spark Labs Academy 데모에서 AI 아바타가 LiveKit 방의 상주 참가자로 보이고, 사용자는 자신의 카메라·마이크와 함께 아바타 영상·음성을 제어합니다. 초기 설계는 오프라인 고품질 영상 생성과 실시간 배포·재생의 조합이며, STT/TTS 기반 실시간 립싱크는 확장 방향입니다.

13. 문서 정책

Markdown과 PDF는 docs-site/public/downloads/에 통합되어 있습니다. 수정 후 npm run sync:docs로 Starlight 페이지에 반영하고 빌드합니다. 상세 규칙은 문서 정책을 참고합니다.