

Design and Implementation of an Emotion-Aware Music Curation Service with a Secure API Proxy Backend

Park Ju Young¹, Choo Yeon Woo², Jeong Jin Myeong³, Kunhee Han^{4*}

¹ Department of Software, Baekseok University, Korea

² Department of Artificial Intelligence, Baekseok University, Korea

³ Department of Multimedia, Baekseok University, Korea

^{4*} Advisor and Professor, Baekseok University, Korea

juwwkd89@gmail.com¹, cndusdn03@bu.ac.kr², 20214760@bu.ac.kr³, hankh@bu.ac.kr^{4*}

Corresponding author*: Kunhee Han

Abstract Music streaming services are ubiquitous, yet finding songs that match a user's current emotional state still depends on manual keyword search and popularity charts that ignore emotional context. Moreover, single-page web clients that call external APIs directly can expose API keys in the browser. This paper presents the design and implementation of an emotion-aware music curation service that addresses both issues. The React frontend sends a user's natural-language description of their mood to a Spring Boot backend, which analyzes the emotion through the Anthropic Claude API and returns matching songs using the YouTube Data API. All external calls are relocated to the backend, which acts as a secure API proxy with response caching, IP-based rate limiting, a global error handler, and an optional Firebase ID-token check. The service is deployed as a single Spring Boot JAR on an Oracle server. Backend tests passed, the frontend production build succeeded, and the deployed health endpoint responded normally. Validation with live API keys and the full login flow remains future work.

Keywords: Emotion analysis, Music recommendation, Large language model, API proxy, Backend security, Spring Boot

1. 서론

음악 스트리밍 서비스가 보편화되었지만, ‘지금의 감정’에 맞는 음악을 찾는 일은 여전히 사용자의 몫이다. 키워드 검색과 인기 차트는 곡의 인기도나 제목·태그에 의존하므로 ‘위로가 필요하다’, ‘집중하고 싶다’와 같이 자연어로 표현된 감정 맥락을 직접 반영하기 어렵다[1]. 또한 단일 페이지 웹 클라이언트가 YouTube·Claude 같은 외부 API를 브라우저에서 직접 호출하면 API 키가 클라이언트 번들에 노출될 수 있다는 보안 문제가 있다.

본 연구는 이 두 문제를 함께 다루는 ‘AI 감정 분석 기반 음악 추천 서비스’를 설계·구현한다. 기여는 다음과 같다. 첫째, 자연어 감정 입력을 대규모 언어 모델(LLM)로 분석해 음악 추천으로 연결하는 흐름을 구현하였다. 둘째, 모든 외부 API 호출을 백엔드로 이전하고 프론트엔드는 자체

백엔드 API만 호출하도록 변경하여 키 노출 위험을 제거하였다. 셋째, 캐시·요청 제한·공통 에러 처리·문서화를 갖춘 Spring Boot 프록시 백엔드를 구축하고 Oracle 서버에 배포해 동작을 확인하였다.

2. 관련연구

음악 추천은 전통적으로 협업 필터링, 콘텐츠 기반 인기도 기반 방식으로 연구되어 왔으며, 사용자가 처한 상황이나 감정 맥락을 반영하기 어렵다는 한계가 지적되어 감정 기반 추천이 사용자 중심 접근으로 주목받아 왔다[1]. 최근에는 자연어로 표현된 감정 상태를 트랜스포머 기반 임베딩으로 분석해 추천에 활용하는 연구도 제안되었다[2]. 본 연구는 별도의 감정 분류 모델을 학습하는 대신 상용 LLM API로 자연어 감정을 분석하고, 이를 안전한 백엔드 프록시 구조와 결합한 점에서 차이가 있다.

3. 시스템 설계

제안 서비스는 프론트엔드, 백엔드, 외부 API의 세 부분으로 구성된다(Figure 1). React·Vite로 구현한 프론트엔드는 음악 추천 UI를 담당하며 외부 API 키를 보유하지 않고 자체 백엔드의 /api/* 엔드포인트만 호출한다. Spring Boot 백엔드는 외부 API 호출을 대리하는 보안 프록시로서, 감정 분석은 Anthropic Claude API[3]로, 차트·검색·재생 정보는 YouTube Data API v3[4]로 중계한다. 로그인과 좋아요·앨범·히스토리 데이터는 프론트엔드 SDK를 통해 Firebase Authentication·Firestore[5]가 담당한다

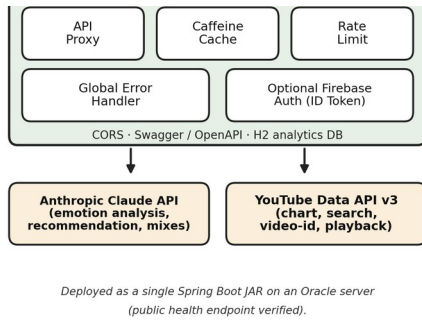


Figure 1. System architecture of the proposed emotion-aware music curation service

사용자가 감정을 자연어로 입력하면 백엔드가 Claude로 감정을 분석하고(POST /api/emotion) 그 결과로 추천 곡을 생성하며(POST /api/recommend), 재생 시점에 곡을 videoId로 변환해(GET /api/video-id) 재생한다. 주요 백엔드 엔드포인트는 Table 1과 같다. 이 외에 맞춤 믹스(/api/mix), 취향 기반 추천 (/api/taste/recommend), 유사 곡 추천(/api/similar-song), 운영 집계(/api/ops/summary)를 포함해 총 11개의 REST API를 구현하였고, 전 API를 Swagger(OpenAPI)로 문서화하였다.

Table 1. Key implemented backend API endpoints

Endpoint	Description
GET /api/health	Service health check
GET /api/chart	Popular music chart (YouTube)
GET /api/search	Music and MV search (YouTube)
GET /api/video-id	Resolve a playable video ID
POST /api/emotion	Emotion analysis (Claude)
POST /api/recommend	Emotion-based recommendation (Claude)
POST /api/mixes	Themed mix generation (Claude)

백엔드는 공통 관심사를 모듈로 분리하였다. CORS 설정과 전역 에러 핸들러로 일관된 응답을 제공하고, Caffeine 캐시로 반복 호출에 따른 외부 API 할당량 소모를 줄이며, 클라이언트 IP 기준 요청 제한으로 과도한 호출을 차단한다. 운영 기록은 H2 데이터베이스에 저장하되 원본 IP·프롬프트 원문·토큰·API 키는 남기지 않고 해시값과 엔드포인트·상태 코드 등 최소 지표만 기록한다. 또한 Firebase ID Token을 검증하는 구조를 선택적으로 적용할 수 있도록 구현하였으며 현재는 기본 비활성 상태이다.

4. 구현 및 검증

구현 스택은 백엔드 Java·Spring Boot·Maven, 프론트엔드 React·Vite이며, 캐시는 Caffeine, 운영 기록은 H2 데이터베이스, API 문서화는

Swagger(OpenAPI), 음악 재생은 YouTube IFrame Player를 사용한다. 배포는 Docker 없이 Vite 빌드 산출물을 Spring Boot 정적 리소스로 내장한 단일 실행 JAR 방식으로, Oracle 서버에서 운영한다.

검증 결과, 백엔드 테스트가 모두 통과하였고(tests run 10, failures 0, errors 0), 프론트엔드 정적 분석(npm run lint)과 프로덕션 빌드(npm run build)가 성공하였으며, Oracle 서버에 배포된 health 엔드포인트가 정상 응답을 반환하였다. 또한 로컬 환경에서 ANTHROPIC_API_KEY가 설정되지 않은 경우 Claude 연동 엔드포인트(예: /api/emotion)가 MISSING_API_KEY 오류를 반환하여 키 보호 구조가 동작함을 확인하였다. 다만 실제 Claude·YouTube API 키를 적용한 추천·검색의 운영 통합 성공 검증과, Firebase 인증 도메인 설정 및 로그인·추천·재생 전체 플로우의 운영 검증은 아직 수행하지 않았으며 이를 향후 과제로 남긴다.

5. 결론

본 연구는 프론트엔드에 노출될 수 있는 외부 API 키 문제를 백엔드 프록시 구조로 개선하고, 외부 AI·데이터 API를 안전하게 연동하는 방법을 정리하였다. 감정 입력에서 추천·재생으로 이어지는 흐름을 구현하는 동시에 캐시·요청 제한·에러 처리·문서화·배포까지 포함한 실무형 백엔드를 구성하였다는 점에서 포트폴리오로서 의미가 있다.

향후에는 Firebase ID Token 검증의 운영 단계 강제화, Redis 기반 캐시 확장, 운영 로그·모니터링 강화, 기록 데이터베이스 확장으로 완성도를 높일 수 있다. 또한 실제 API 키 기반 운영 통합 검증과 추천 품질의 정량적 평가도 함께 진행할 계획이다.

References

- [1] Song, Y., Dixon, S., & Pearce, M. (2012). A survey of music recommendation systems and future perspectives. In *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)* (pp. 395–410).
- [2] Chavali, A., & Menezes, R. (2025). *Words to waves: Emotion-adaptive music recommendation system* (arXiv:2510.21724). arXiv. <https://arxiv.org/abs/2510.21724>
- [3] Anthropic. (n.d.). *Claude API documentation*. Retrieved June 5, 2026, from <https://platform.claude.com/docs>
- [4] Google. (2026). *YouTube Data API v3*. Google for Developers. <https://developers.google.com/youtube/v3>
- [5] Google. (2026). *Firestore Authentication documentation*. Firebase. <https://firebase.google.com/docs/auth>