

API 명세서_박진양 (ver.2)

1. 인증 (AUTH)

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-AUTH-002</u>	POST	/auth/verify-email	이메일 인증 코드 검증	<ul style="list-style-type: none">email (string, required): 회원 이메일token (string, required): 6자리 인증 코드	<pre>{ "message": "이메일 인증이 완료되었습니다." }</pre>
<u>API-AUTH-003</u>	POST	/auth/login	로그인 (JWT/세션 발급)	<ul style="list-style-type: none">email (string, required): 회원 이메일password (string, required): 비밀번호	<pre>{ "message": "로그인 성공", "user_id": 1 }</pre>
<u>API-AUTH-004</u>	POST	/auth/logout	로그아웃 (세션/토큰 무효화)	-	<pre>{ "message": "로그아웃 되었습니다." }</pre>
<u>API-AUTH-005</u>	POST	/auth/find-email	이름·회사명으로 가입 이메일 조회	<ul style="list-style-type: none">name (string, required): 사용자 이름company_name (string, required): 회사명	<pre>{ "email": "jo****@example.com" }</pre>
<u>API-AUTH-006</u>	POST	/auth/request-password-reset	비밀번호 재설정 코드 발송	<ul style="list-style-type: none">email (string, required): 회원 이메일	<pre>{ "message": "재설정 코드를 이메일로 전송했습니다." }</pre>

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-AUTH-007</u>	POST	/auth/verify-password-reset-code	비밀번호 재설정 코드 검증	<ul style="list-style-type: none"> email (string, required): 회원 이메일 code (string, required): 6자리 인증 코드 	{ "message": "인증이 완료되었습니다." }
<u>API-AUTH-008</u>	POST	/auth/reset-password	검증된 코드로 비밀번호 변경	<ul style="list-style-type: none"> email (string, required): 회원 이메일 code (string, required): 인증 코드 new_password (string, required): 새 비밀번호 (8~20자) 	{ "message": "비밀번호가 재설정되었습니다." }
<u>API-AUTH-009</u>	GET	/auth/me	내 계정 정보 조회	Headers: Cookie: user_id=<id> 또는 Authorization: Bearer <token>	<pre>{ "id": 1, "email": "user@example.com", "name": "홍길동", "company": "A사", "is_verified": true }</pre>

2. 이메일 (Email)

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-EMAIL-001</u>	POST	/email/send-code	인증 코드 생성. 이메일 발송	<ul style="list-style-type: none"> email (string, required): 회원 이메일 	{ "message": "인증 코드가 이메일로 전송되었습니다." }
<u>API-EMAIL-002</u>	POST	/email/verify-code	이메일 인증 코드 검증	<ul style="list-style-type: none"> email (string, required) code (string, required) 	{ "message": "이메일 인증이 완료되었습니다." }

Aa API ID	📌 HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-EMAIL-003</u>	POST	/email/register-after-verification	이메일 인증 후 회원가입	<ul style="list-style-type: none"> • email (string, required): 인증이 완료된 이메일 주소 • password (string, required, 8~20자): 사용할 비밀번호 • name (string, required): 사용자 이름 • company (string, required): 회사명 	{ "message": "회원가입이 완료되었습니다." }
<u>API-EMAIL-004</u>	GET	/email/check-verified	이메일 인증 상태 조회	<ul style="list-style-type: none"> • Query: email (string, required): 인증 상태를 확인할 이메일 주소 	{ "verified": true }

3. 챗봇 (Chat)

Aa API ID	📌 HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-CHAT-001</u>	GET	/chat/{user_id}	특정 사용자의 대화 이력 조회	Path: user_id (integer, required)	List of ChatHistoryResponse: <ul style="list-style-type: none"> • id • user_id • question • response • created_at

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-CHAT-002</u>	POST	/chat	질문 전송 → RAG 파이프라인 실행 → 응답 저장	<ul style="list-style-type: none"> chatroom_id (integer, optional) question (string, required) 	ChatResponse: <ul style="list-style-type: none"> id response created_at

4. 채팅방 (ChatRoom)

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-CHATROOM-001</u>	POST	/chat_rooms/	새 채팅방 생성	<ul style="list-style-type: none"> title (string, required): 채팅방 제목 	ChatRoomOut: <ul style="list-style-type: none"> id user_id title created_at
<u>API-CHATROOM-002</u>	GET	/chat_rooms/list	내 채팅방 목록 조회	Headers: Authorization	List of ChatRoomOut: <ul style="list-style-type: none"> id user_id title created_at
<u>API-CHATROOM-003</u>	GET	/chat_rooms/{room_id}	채팅방 상세 조회	Headers: Authorization Path: room_id (int, required)	ChatRoomOut: <ul style="list-style-type: none"> id user_id title created_at
<u>API-CHATROOM-</u>	GET	/chat_rooms/{room_id}/messages	채팅	Headers: Authorization	List of ChatHistoryResponse:

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>004</u>			방 내 메시지 조회	Path: <code>room_id</code> (int)	<ul style="list-style-type: none"> <code>id</code> <code>chatroom_id</code> <code>user_id</code> <code>question</code> <code>response</code> <code>created_at</code> <code>feedback_given</code>

5. 피드백 (Feedback)

Aa API ID	⌵ HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API- FEEDBACK- 001</u>	POST	<code>/feedback</code>	채팅 피드백 제출	<ul style="list-style-type: none"> <code>chat_id</code> (integer, required): 대상 채팅 ID <code>score</code> (integer, required): (0: 나쁨 / 1: 좋음) 	FeedbackResponse: <ul style="list-style-type: none"> <code>id</code> <code>chat_id</code> <code>score</code> <code>created_at</code>
<u>API- FEEDBACK- 002</u>	GET	<code>/feedback/all</code>	(관리자) 피드백 전체 조회	Headers: <code>Authorization</code> (admin only)	List of FeedbackResponse: <ul style="list-style-type: none"> <code>id</code> <code>chat_id</code> <code>score</code> <code>created_at</code>

6. 사용자 정보 · 마이페이지 (Users)

Aa API ID	🔍 HTTP Method	≡ API Path	≡ 기능	≡ Request schema	≡ Response schema
<u>API-USERS-001</u>	GET	/users/me	내 프로필 조회	Headers: Authorization	UserOut: <ul style="list-style-type: none"> id email name company is_verified
<u>API-USERS-002</u>	PUT	/users/me	프로필 수정	<ul style="list-style-type: none"> name (string, optional): 새 이름 company (string, optional): 새 회사명 password (string, optional): 새 비밀번호 (8자~) 	UserOut: <ul style="list-style-type: none"> id email name company is_verified updated_at
<u>API-USERS-003</u>	PUT	/users/me/password	비밀번호 변경	<ul style="list-style-type: none"> current_password (string, required): 기존 비밀번호 new_password (string, required): 새 비밀번호 (8자 ~) 	{ "message": "비밀번호가 변경되었습니다." }