</> AI활용 웹 개발 세미프로젝트 API 명세서(Nody Chat)

All Documentation | By Section | Endpoints List

Router	Title	Section	Method	Endpoint	설명	인증 여부	Request Body	Responce	Status Code	추가 설명
API Overview	API 개요	Overview		N/A	WebChatting API	☐				Title에서 사이드에서 보기 열어서 내용 확인
	</> 로그인 페이지	Endpoints	GET	/	로그인 페이지를 불러옵니다.	☐		res.clearCookie("token") 200 : OK { "Conte	200 : OK 500 : Server Error	클라이언트가 처음 방문하는 진입점 페이지 로그인 페이지를 불러올 때 쿠키 삭제
User	</> 로그인	Endpoints	POST	/user/login	사용자가 아이디와 비밀번호로 로그인합니다.	☑	`{ "userid": "apple", "passwd": "11:`	200: OK 토큰 인증 성공시 res.redirect("/mai	200 : OK 404 : Not Found 400 : Bad Request 500 : Server Error	인증에 성공하면 JWT 토큰이 쿠키에 저장되고, `/main` 페이지로 리디렉션 됩니다.
	</> 유저 아이디와 닉네임 매핑	Endpoints	GET	/user/user-map	아이디와 닉네임을 매핑 시켜데이터베이스에서 정보를 가져오기 위한 기능	☐	{ "nicknames" : ["user.nickname1", "us	200 : OK { "userid" : "nickname1", "userid" :	500 : Server Error 200 : OK	페이지에선 닉네임으로 사용되기 때문에 닉네임과 아이디를 매핑시켜 데이터베이스에 아이디로 정보를 요청하고 저장
	</> 닉네임과 유저 아이디 매핑	Endpoints	POST	/user/nicknames-to-ids	닉네임과 아이디를 매핑시켜 아이디 값에 맞는 정보를 데이터베이스에 보내기 위한 기능	☐	{ "friend" : ["user.nickname1", "user.nic	200 : OK { "userids": ["user.nickname1", "us	200 : OK 400 : Bad Request 500 : Server Error	반대로 데이터를 데이터베이스에 보내기 위해서 페이지의 닉네임과 매핑되는 아이디 값을 받아 데이터베이스와 연결
	</> 로그인한 사용자의 친구추가	Endpoints	POST	/user/add-friend	친구 추가 기능을 제공합니다.	☑	`{ "newNickname": "newName" }`	200 : OK { "message": "newName님을 친구	200 : OK 400 : Bad Request 404 : Not Found 409 : Conflict	친구 검색 란에 닉네임을 입력하면 같은 입력 값의 유저가 출력 되고 추가 버튼 클릭 시 친구 추가가 되고 친구 목록에 추가 합니다.
	</> 로그인한 사용자 개인정보 보내주기	Endpoints	GET	/user/me	마이페이지에 들어갈 회원 정보를 데이터베이스에서 가져옵니다.	☑	`{ "userid": "apple" }`	200 : OK `{ "userid": "apple", "email":`	200 : OK 404 : Not Found	
	</> 로그인한 사용자의 친구 목록 보내주기	Endpoints	GET	/user/get-friend	유저 아이디에 저장된 친구 목록을 데이터베이스에서 가져옵니다.	☑	`{ "userid": "apple" }`	200 : OK { "friend" : ["banana", "melon"] } 40	200 : OK 404 : Not Found	
	</> 로그인한 사용자의 친구 삭제	Endpoints	POST	/user/remove-friend	친구 목록에 있는 유저를 삭제하는 기능	☑	{ "nickname": "멜론" }	200 : OK { "message": "멜론님이 친구 목록에	200 : OK 400 : Bad Request 404 : Not Found	
	</> 회원가입 페이지	Endpoints	GET	/signup	회원가입 페이지를 불러옵니다.	☐		200 : OK { "Content-Type": "text/html" } 50	200 : OK 500 : Server Error	
	</> 회원가입	Endpoints	POST	/user/signup	새 사용자를 생성합니다. 회원가입 시 필수 입력값이 같은 유저가 있는지 확인합니다.	☑	`{ "nickname": "김사과", "email": "a`	201: created `{ "token": "jwt_token_strin`	201 : Created 400 : Bad Request 409 : Conflict 500 : Server Error	이메일, 전화번호, 닉네임, 아이디는 중복될 수 없습니다.
	</> 아이디 찾기	Endpoints	POST	/user/findID	회원가입 시 입력했던 이메일로 아이디를 찾습니다.	☐	`{ "email": "apple@apple.com" }`	200: OK `{ "userid": "apple" }` 401 : Una	200 : OK 401 : Unauthorized	
	</> 회원가입용 인증번호 보내기	Endpoints	POST	/user/send-signupCod	회원가입 시 유저가 입력한 전화번호로 문자 인증 번호를 보내는 기능을 합니다.	☐	`{ "phone" : "01011111111" }`	`{ "message": "인증번호 전송 완료", "sess`	200 : OK 500 : Server Error	외부 API인 coolsms를 활용했습니다. 인증 번호의 유효 시간은 3분으로 설정, uuid로 생성한 세션 키를 폰 번호와 매칭시켜 이후 인증 시 세션 키를 활용
	</> 인증번호 확인	Endpoints	POST	/user/verify-code	보낸 인증번호와 입력한 인증번호가 같은지 비교합니다.	☑	{ "sessionId" : "sessionkey", "code" : "0	400 : Bad Request { "message": "유효하지 않	200 : OK 400 : Bad Request 401 : Unauthorized	Coolsms 외부 API를 활용, 입력한 번호로 메세지를 전송하고 유저는 그 코드를 입력하여 값 비교, 일치 시 인증 성공 인증 번호의 유효 시간은 3분으로 설정
	</> 회원정보 수정	Endpoints	POST	/user/update	이메일, 닉네임, 전화번호, 비밀번호 등 회원 정보를 선택적으로 수정할 수 있습니다.	☑	`{ "newNickname": "newName", "newEma`	200: OK `{ "message": "회원정보가 성공적으`	200 : OK 400 : Bad Request 404 : Not Found 409 : Conflict 500 : Server Error	
	</> 비밀번호 재설정용 인증번호	Endpoints	POST	/user/send-code	회원 가입에 작성한 전화번호에 문자로 인증 번호를 전송합니다.	☐	`{ "phone" : "01011111111" }`	200: OK `{ "message": "인증번호 전송 완 료`	200 : OK 409 : Conflict 500 : Server Error	외부 API인 coolsms를 활용했습니다. 인증 번호의 유효 시간은 3분으로 설정, uuid로 생성한 세션 키를 폰 번호와 매칭시켜 이후 인증 시 세션 키를 활용 전화 번호를 입력 받아 기존 유저 중에 같은 번호가 있는지 비교하고 없으면 오류 메세지 출력, 있으면 고유 세션 키를 생성하여 번호와 매칭시켜 저장, 인증 번호를 보내 유저가 입력하면 같은지 비교한다.
	</> 비밀번호 변경	Endpoints	POST	/user/reset-password	인증을 통해 비밀번호를 변경합니다.	☑	{ "sessionId" : "sessionkey", "newPassc	403 : Forbidden { "message": "세션이 만료도	200 : OK 403 : Forbidden 404 : Not Found	전화 번호와 매핑된 세션 아이디 값을 확인하여 인증 진행 후 그 결과 값에 따라 새로운 비밀번호 생성 여부를 판단합니다
main	</> 메인 채팅 페이지	Endpoints	GET	/main	인증된 사용자에게 보여지는 메인 페이지입니다.	☑		"./mainPage/mainpage.html" {"message" : "	200 : OK 500 : Server Error	메인페이지 HTML를 반환.
ChatRoom	채팅방 생성	Endpoints	POST	/chatRoom/create	새로운 채팅방을 생성합니다.	☑	`{ "name": "채팅방", "members": ["us`	201 : Created {"newRoom": "name"} 404 : N	201 : Created 400 : Bad Request 404 : Not Found 500 : Server Error	요청자는 자동으로 멤버에 포함됩니다.
	</> 채팅방 목록 조회	Endpoints	GET	/chatRoom/get	로그인한 사용자가 참여 중인 채팅방 목록을 반환합니다.	☑	`{ "id" : "_id", "roomId": "채팅방 I`	200 : OK `{ "members": ["user1", "user2",`	200 : OK 404 : Not Found 500 : Server Error	
	</> 채팅방 참가자 조회	Endpoints	GET	/chatRoom/members?r	채팅방의 참여자 목록을 채팅방 헤더에 반환합니다.	☐	{ "chatroomId" : "roomid", "members"	200 : OK `{ "members": ["user1", "user2",`	200 : OK 404 : Not Found 500 : Server Error	
	</> 채팅방 초대	Endpoints	POST	/chatRoom/invite	기존 채팅방에 친구들을 초대합니다.	☑	`{ "id" : "_id" "roomId": "채팅방 ID`	200 : OK `{ "message": "2명 초대 완료", "`	200 : OK 400 : Bad Request 403 : Forbidden 404 : Not Found 500 : Server Error	초대자는 채팅방의 기존 멤버여야 합니다.
	</> 채팅방 나가기	Endpoints	POST	/chatRoom/leave	사용자가 채팅방에서 나갑니다.	☑	`{ "id" : "_id", "roomId": "662d2e8`	200 : OK `{ "message": "채팅방이 삭제되었습`	200 : OK 400 : Bad Request 403 : Forbidden 404 : Not Found 500 : Server Error	본인이 나간 후 채팅방 멤버가 한명 이하일 경우 채팅방이 자동으로 삭제, 2명 이상일 경우 채팅방 멤버 업데이트
Message	</> 채팅 보내기	Endpoints	POST	/message/send	입장한 채팅방에서 입력란에 타이핑 한 내용을 실시간 채팅창에 출력합니다.	☑	`{ "id" : "_id", "chatroomId": "662`	res.status(201).json(saved) 예시) { "_id": "m	200 : OK 400 : Bad Request 500 : Server Error	
	</> 채팅 불러오기	Endpoints	POST	/message/load	각 채팅방에서 입력한 채팅 내역을 채팅방 입장때 마다 자동으로 불러옵니다.	☑	`{ "id" : "_id", "chatroomId": "662`	200 : OK `{ "chatroomId": "662d2e8f05dd4`	200 : OK 404 : Not Found 403 : Forbidden 500 : Server Error	
newsParsing	</> 기사 크롤링	Endpoints	GET	/news	KBS 랭킹뉴스 5위까지의 랭킹순위, 제목, 링크를 크롤링해서 가져옵니다.	☑		200 : OK { "num" : "1", "headline" : "한달 등록	200 : OK 500 : Server Error	