

DTO

name	attributes
UserDto	<pre>{ id: Long, username: String, profileImage: String, name: String, nickname: String, birthYear: String, phone: String, registerDate: String, isAdmin: Boolean, gender: Integer, isBanned: Boolean }</pre> <p>유저의 간단한 정보(프로필 이미지 주소, nickname만)만 받을 수 있는 api를 만들고, 유저 상세 조회는 본인만 받을 수 있도록 설정해주세요.</p>
TokenDto	<pre>{ 토큰 관련 전부 }</pre>
ProfileDto	<pre>{ userId: Long, profileImage: String, nickname: String }</pre>
UpdateUserDto	<pre>{ userPassword: String, userPhone: String }</pre>
ProductDto	<pre>{ productId: Integer, productType: String, regionId: String, user: IProductUser{ userId, userEmail, profileImage, nickname, isBanned }, productAddress: String, productDeposit: Integer, productRent: Integer, productMaintenance: Integer, }</pre>

	<pre> productMaintenanceInfo: String, productIsSupportable: Boolean, productIsFurnitureSupportable: Boolean, productSquare: Float, productRoom: Integer, productOption: String, productAdditionalOption: String, productIsBanned: Boolean, productIsDeleted: Boolean, productPostDate: String, productStartDate: String, productEndDate: String, lat: double, lng: double, productAddressDetail: String } </pre>
ProductReturnDto	<pre> { productId: Integer, productType: String, region: Region{ regionId: String, regionSido: String, regionDong: String, regionGugun: String, }, IProductUser: IProductUser{ userId, profileImage, nickname, IsBanned }, productAddress: String, productDeposit: Integer, productRent: Integer, productMaintenance: Integer, productMaintenanceInfo: String, productIsSupportable: Boolean, productIsFurnitureSupportable: Boolean, productSquare: Float, productRoom: Integer, productOption: Integer, productAdditionalOption: List<String> productIsBanned: Boolean, productPostDate: String, productStartDate: String, productEndDate: String, lat: double, lng: double, productAddressDetail: String, productIsInterest: Boolean mediaList: List<Media> } </pre>
ProductUpdateDto	<pre> { productDeposit: Integer, productMaintenance: Integer, </pre>

	<pre> productIsRentSupportable: Boolean, productIsFurnitureSupportable: Boolean, productOption: String, productAdditionalOption: String, productStartDate: Date, productEndDate: Date } </pre>
ProductUploadDto	<pre> { productType: String, regionId: String, productAddress: String, productDeposit: Integer, productRent: Integer, productMaintenance: int, productMaintenanceInfo: String, productIsRentSupportable: Boolean, productIsFurnitureSupportable: Boolean, productSquare: Float, productRoom: Integer, productOption: Integer, productAdditionalOption: List<String>, productPostDate: Date, productStartDate: Date, productEndDate: Date, lat: Float, lng: Float, } </pre>
ProductSearchDto	<pre> { } </pre>
PreferenceDto	<pre> { preferenceId: Long, preferenceName: String, userId: Long, regionId: String, preferenceDeposit: Integer, preferenceRent: Integer, preferenceType: Integer, preferenceInfra: String, preferenceStartDate: String, preferenceEndDate: String } </pre>
ChatroomDto	<pre> { chatroomId: Long, chatroomtitle: String, productId: Integer, creator: Integer, (매물 게시물 만든사람) participant: Integer, (매물에 관심있는 사람) } </pre>

	ChatroomCreatedAt: String, chatroomParticipantIsOut: Boolean chatroomCreatorIsOut:Boolean }
ChatroomReturnDto	{ chatroomId: Long, product: ProductDto, user: UserDto, }
ChatDto	{ chatId: Integer, chatroomId: Integer, chatContent: String, receiver: User, chatTime: String, mediaList: List<Media> }
ChatReturnDto	{ chatId: Integer, chatroomId: Integer, chatContent: String, receiver: ProfileDto,, chatTime: String, mediaList: List<Media> }
BoardDto	{ boardId: Integer, boardTitle: String, boardContent: String, boardWriter: IUser, boardRegion: Region boardHit: Integer boardDate: Date, boardIsBanned: Boolean boardIsDelete: Boolean, boardHit:int }
BoardReturnDto	{ boardId: Integer, boardTitle: String, boardContent: String, boardWriter: IUser, region: Region, boardHit: Integer, boardDate: Date, boardIsBanned: Boolean }

IUser	{ userId: Long, nickname: String, isBanned: Boolean }
CommentDto	{ id: Long, nickname: String, isBanned:boolean, content: String, date: String, subComments: List<SubComment> }
AlarmMessageDto	{ alarmMessageId:Long, userId: Long, alarmMessageDate: Date, alarmMessage: String }
AlarmSettingDto	{ alarmId: Long, userId: Long, alarmPhoneChat: Boolean, alarmPhoneComplete: Boolean, alarmPhoneInterest: Boolean, alarmEmailChat: Boolean, alarmEmailInterest: Boolean, alarmEmailComplete: Boolean }
ReportDto	{ reportSubject: Integer, reportTypeId: Integer, content: String, subjectId: Long }
ScheduleDto	{ scheduleId: Long, chatroomId: Long, scheduleDate: Date, }
RegionSidoDto	{ regionId: String, regionSido: String }
RegionGugunDto	{ regionId: String,

	regionSido: String }
RegionDongDto	{ regionId: String, regionSido: String }
UserInformationDto	{ userId: Long, email: String, password: String, nickname: String, profileImage: String, phone: String }

API 명세서 (“/api”)

상위 기능	하위 기능	REST	파라미터	반환
회원 관리 (/users)	회원가입 (/register)	POST	{ email: String, password: String, name: String, birthYear: String, phone: String }	{ status: 200, message: SUCCESS }
<회원 DB 수정> 회원가입 회원탈퇴	로그인 (/login)	POST	{ email: String, password: String, sso: String }	{ status: 200, message: SUCCESS, user: userDto }
<인증> 로그인 로그아웃 아이디 찾기 비밀번호 찾기	로그아웃 (/logout)	PUT	-	{ status: 200, message: SUCCESS }
<프로필, 정보 관리> 내 프로필 조회	내 프로필 조회 (/profile/:userId)	GET	-	{ status: 200, }

상위 기능	하위 기능	REST	파라미터	반환
프로필 수정 내 회원 정보 조회 회원 정보 수정				message: SUCCESS, profile: profileDto }
	내 회원 정보 조회 (/:userId)	GET	-	{ status: 200, message: SUCCESS, user: userInformationDto }
	회원 정보 수정 (/modify/:userId)	POST	{ email: String, password: String, phone: String }	{ status: 200, message: SUCCESS }
	프로필 수정 (/modify/:userId/profile)	PUT	{ nickname: String, profileImage: MultipartFile, }	{ status: 200, message: SUCCESS, profile: profileDto }
	회원탈퇴 (/delete)	DELETE	-	{ status: 200, message: SUCCESS }
	아이디 찾기 (/find/id)	POST	{ phone: String }	{ status: 200, message: SUCCESS, email: String }
	비밀번호 확인 (/password)	POST	{ userId: Long, password: String }	{ status: 200, message: SUCCESS, data: boolean (true / false) }
	다크 모드 설정 (/dark)	GET	-	{ status: 200, message: SUCCESS }
	소셜 로그인: 구글 (/oauth2/authorization/google)	GET	-	{ status: 200, message: SUCCESS }

상위 기능	하위 기능	REST	파라미터	반환
				}
	소셜 로그인: 네이버 (/oauth2/authorization/naver)	GET	-	{ status: 200, message: SUCCESS }
	소셜 로그인: 카카오 (/oauth2/authorization/kakao)	GET	-	{ status: 200, message: SUCCESS }
이메일 인증 (/emails)	이메일 인증 번호 발송	POST	{ email: String }	{ status: 200, message: SUCCESS }
	인증 번호 확인 (/verify)	POST	{ email: String, code: String }	{ status: 200, message: SUCCESS, result: (boolean) true / false }
	비밀번호 찾기: 인증 번호 확인 (/find/password)	POST	{ email: String, code: String }	{ status: 200, message: SUCCESS, result: (boolean) true / false }
매물 관리 (/products)	매물 업로드 (/upload)	POST	{ productUploadDto: { type: String, regionId: String, address: String, deposit: Integer, rent: Integer, maintenance: Integer, maintenanceInfo: String, isRentSupportable: Boolean, isFurnitureSupportable: Boolean, square: float, room: Integer, option: Integer, additionalOption: String, startDate: Date, }	{ status: 200, message: SUCCESS }

상위 기능	하위 기능	REST	파라미터	반환
			endDate: Date, AddressDetail: String, lat: Double, lng: Double, } productMedia: List<MultipartFile> }	
<매물 편집> 매물 업로드 매물 수정 매물 삭제	매물 수정 (/modify/:productId)	PUT	{ productDeposit: Integer, productRent: Integer, productMaintenance: Integer, productIsRentSupportable: Boolean, productIsFurnitureSupportable: Boolean, productOption: Integer, productAdditionalOption: String, productStartDate: String, productEndDate: String, }	{ status: 200, message: SUCCESS, product: productDto }
<매물 조회> 매물 조회 매물 검색	매물 삭제 (/delete/:productId)	DELETE	-	{ status: 200, message: SUCCESS }
	매물 조회 (/:productId)	GET	-	{ status: 200, message: SUCCESS, product: productDto }
	매물 검색 (/search)	POST	{ order: Integer, minDeposit: Integer, maxDeposit: Integer, minRent: Integer, maxRent: Integer, type: String, regionId: String, rentSupportable: Boolean, furnitureSupportable: Boolean, infra: Integer, (편의점 등등) startDate: String, endDate: String, lat: float, }	{ status: 200, message: SUCCESS, product: List<ProductDto> }

상위 기능	하위 기능	REST	파라미터	반환
			lng: float }	
	관심 지역의 최신 매물 조회 (/recent/{userId})	GET		{ status: 200, message: SUCCESS, product: ProductDto }
	매물 개수 조회 (/count)	GET	-	{ status: 200, message: SUCCESS, size: Integer }
	지역 최근 매물 불러오기 (/recent/product) (로그인 되어 있을때만 사용가능)	GET	-	{ status: 200, message: SUCCESS, productReturnDto: ProductReturnDto }
	관심 채팅, 관심 매물 불러오기 (/interest/{productId})	GET	-	{ status: 200, message: SUCCESS, data:{ interestCnt: Long, chatCnt: Long } }
	판매자 게시물 불러오기 (/myproducts)	GET	-	{ status: 200, message: SUCCESS, data:{ List< ProductReturnDto> } }
	선호 지역 매물 가져오기 (/preference/{preferenceId})	GET	-	{ status: 200, message: SUCCESS, data:{ List< ProductReturnDto> } }

상위 기능	하위 기능	REST	파라미터	반환
관심 매물 관리 (/interests)	관심 매물 등록 (/add)	POST	{ userId: Integer, productId: Integer }	{ status: 200, message: SUCCESS }
	관심 매물 삭제 (/delete/:userId/:productId)	DELETE	-	{ status: 200, message: SUCCESS }
	관심 매물 조회 (/userId)	GET	-	{ status: 200, message: SUCCESS, product: List<ProductDto> }
선호 설정 관리 (/preferences)	선호 설정 등록 (/add/:userId)	POST	{ PreferenceName: String, regionAddress: String, PreferenceDeposit: Integer, PreferenceRent: Integer, PreferenceType: Integer, PreferenceInfra: Integer, PreferenceStartDate: Date, PreferenceEndDate: Date }	{ status: 200, message: SUCCESS }
	선호 설정 수정 (/modify/:preferenceId)	PUT	{ PreferenceName: String, region: String, PreferenceDeposit: Integer, PreferenceRent: Integer, PreferenceType: Integer, PreferenceInfra: Integer, PreferenceStartDate: Date, PreferenceEndDate: Date }	{ status: 200, message: SUCCESS, preference: preferenceDto }
	선호 설정 삭제 (/delete/:preferenceId)	DELETE	-	{ status: 200, message: SUCCESS }
	선호 설정 불러오기 (/preferenceId)	GET	-	{ status: 200, message: SUCCESS, }

상위 기능	하위 기능	REST	파라미터	반환
				<pre> preference: preferenceDto } </pre>
	선호 설정 목록 조회(/:userId/list)	GET	-	<pre> { status: 200, message: SUCCESS, preference: List<preferenceDto> } </pre>
채팅방 관리(/chatrooms) <채팅방> 채팅방 만들기 채팅방 종료 채팅방 목록 불러오기	채팅방 만들기(/save)	POST	<pre> { title: String, maker: Long, participant: Long, productId: Long } </pre>	<pre> { status: 200, chatroomId: Long, message: SUCCESS } </pre>
	채팅방 나가기 (/end/:chatroomId/:userId)	PUT		<pre> { status: 200, message: SUCCESS } </pre>
	채팅방 목록 불러오기 (/:userId)	GET	-	<pre> { status: 200, message: SUCCESS, chatroom: List<ChatroomReturnDto> } </pre>
	채팅 불러오기 (/chat/:chatroomId)	GET	-	<pre> { status: 200, message: SUCCESS, chat: List<ChatReturnDto> } </pre>
	채팅 로그(/out/{chatroomId})	get	-	
채팅 관리 (/chats) <채팅> 채팅 저장 채팅 불러오기	채팅 저장 (/save)	POST	<pre> { sender: Long, receiver: Long, content: String media: List<MultipartFile> } </pre>	<pre> { status: 200, message: SUCCESS } </pre>
게시판 관리 (/boards)	게시글 작성 (/write/:regionId) regionId: nullable	POST	<pre> { boardTitle: String, boardContent: String, } </pre>	<pre> { status: 200, message: SUCCESS } </pre>

상위 기능	하위 기능	REST	파라미터	반환
			boardWriter: Integer, }	}
<게시글 편집> 게시글 작성 게시글 수정 게시글 삭제	게시글 수정 (/modify/:boardId)	PUT	{ boardTitle: String, boardContent: String }	{ status: 200, message: SUCCESS, board: BoardResultDto }
<게시글 조회> 개별 게시글 조회 게시글 목록 조회 게시글 검색	게시글 삭제 (/delete/:boardId)	PUT	-	{ status: 200, message: SUCCESS }
	개별 게시글 조회 (/:boardId)	GET	-	{ status: 200, message: SUCCESS, board: BoardDto }
	게시글 목록 조회 (/list)	POST	{ regionId: String, pageNo: String, size: String keyword: String }	{ status: 200, message: SUCCESS, board: List<BoardDto> }
지역 검색 (/regions)	도시 검색 (/)	GET	-	{ status: 200, message: SUCCESS, region: RegionSidoDto }
	시군구명 검색 (/:sido)	GET	-	{ status: 200, message: SUCCESS, region: RegionGugunDto }
	읍면동명 검색 (/gugun/:gugun)	GET	-	{ status: 200, message: SUCCESS, region: RegionDongDto }

상위 기능	하위 기능	REST	파라미터	반환
	지역 Id로 지역 시군구 검색 (/:regionId)	GET	-	{ status: 200, message: SUCCESS, region: RegionReturnDto{ regionId: String, regionSido: String, regionDong: String, regionGugun: String, }, }
댓글 관리 (/comments)	댓글 작성 (/:boardId/write)	POST	{ content: String, parentId: Integer }	{ status: 200, message: SUCCESS }
	댓글 수정 (/modify/:commentId)	PUT	{ content: String }	{ status: 200, message: SUCCESS, board: CommentDto }
	댓글 삭제 (/delete/:commentId)	DELETE	-	{ status: 200, message: SUCCESS }
	댓글 목록 조회 (/:boardId)	POST	-	{ status: 200, message: SUCCESS, comment: List<CommentDto> }
신고 관리 (/reports)	신고 전송	POST	{ reportSubjectTypeId: Integer, reportTypeId: Integer, content: String, subjectId: Integer } (* reportSubjectTypeId: 신고 대상의 유형 / 게시글: 0, 매물: 1, 사용자: 2) (* reportTypeId: 신고 이유 유형 / 유형 미정)	{ status: 200, message: SUCCESS }
알림 관리 (/alarms)	알림 권한 설정 (/setting/modify/:userId)	PUT	{ phoneChat: Boolean, }	{ status: 200, }

상위 기능	하위 기능	REST	파라미터	반환
			phoneComplete: Boolean, phoneInterest: Boolean, emailChat: Boolean, emailComplete: Boolean, emailInterest: Boolean }	message: SUCCESS }
<알림 권한 관리> 알림 권한 설정 알림 권한 조회	알림 권한 조회 (/setting/:userId)	GET	-	{ status: 200, message: SUCCESS, alarm: List<AlarmSettingDto> }
<알림 관리> 알림 목록 조회 알림 삭제 알림 생성	알림 목록 조회 (/userId/list)	GET	-	{ status: 200, message: SUCCESS, alarm: List<AlarmMessageDto> }
	알림 삭제 (/delete/:alarmMessageId)	DELETE	-	{ status: 200, message: SUCCESS }
	알림 생성	POST	{ userId: Long, alarmMessage: String }	{ status: 200, message: SUCCESS }
예약 관리 (/schedules)	예약 생성 (/add)	POST	{ chatroomId: Long, date: String }	{ status: 200, message: SUCCESS }
<예약 편집> 예약 생성 예약 삭제 예약 편집	예약 삭제 (/delete/:scheduleId)	DELETE	-	{ status: 200, message: SUCCESS }
<예약 조회> 예약 목록 조회 개별 예약 상세 조회	예약 목록 조회 (/list/:chatroomId)	GET	-	{ status: 200, message: SUCCESS schedule: List<ScheduleDto> }
	개별 예약 상세 조회 (/schedule/:scheduleId)	GET	-	{ status: 200, }

상위 기능	하위 기능	REST	파라미터	반환
				message: SUCCESS schedule: ScheduleDto }
	예약 편집 (/modify/:scheduleId)	PUT	{ date: String }	{ status: 200, message: SUCCESS schedule: ScheduleDto }
파일 관리 (/files)	파일 올리기	POST	{ files: List<Multipart> }	{ status: 200, message: SUCCESS }
<파일 관리> 파일 올리기	파일 가져오기	GET	{ }	

오류 코드

- 200 + message