

Domain	Method	URL	URL Params	Data Params	Response	Description	Authority
Auth	POST	/api/login		{ nickname : String, password : String }	200 Header – Authorization : Access Token, Header - Cookie : Refresh Token	로그인	
	GET	/api/logout			200	로그아웃	
	GET	/api/access			200	토큰 재발급	
Mail	POST	/api/mail/request		{ email : String }	200 email 로 메일 전송	메일 인증 번호 발송 (아이디 찾기 또는 비밀번호 찾기 시)	
	POST	/api/mail/sign		{ email : String }	200 email 로 메일 전송 201 중복된 이메일 존재 시 메일 전송하지 않음	메일 인증 번호 발송 (회원가입 시)	
	POST	/api/mail/validate		{ email : String, authCode : String }	200 인증 완료 201 인증 실패	메일 인증 확인	
	POST	/api/mail/find		{ email : String, authCode : String }	200 { username : String }	아이디 찾기 (앞의 두 글자만 공개)	
	POST	/api/mail/reset		{ username : String, password : String }	200 완료 문구 : String	비밀번호 재설정	

User	POST	/api/users/signup		{ name : String, nickname : String, username : String, password : String, phone : String, email : String }	200 회원 가입 완료 : String	회원가입	
	GET	/api/users/check		{ username : String }	200 { username : String } 404	아이디 중복 확인	
	GET	/api/users/preview/{tag}	{ tag : String }		200 { nickname : String, profile : String, star : double, tag : String } 404	선택한 사용자 정보 조회	
	GET	/api/users/read			200 { name : String, nickname : String, username : String, phone : String, email : String, profile : String }	본인의 상세 정보 조회	USER

					404		
	PATCH	/api/users/update		Data : { name : String, nickname : String, username : String, password : String, phone : String, email : String }, Image : MultipartFile	200 수정 완료 : String 404	프로필 수정	USER
	PATCH	/api/users/change		{ word : String }	200 수정 완료 : String 404	비밀번호 변경	USER
	DELETE	/api/users/delete		{ word : String }	200 탈퇴 완료 : String	회원 탈퇴	USER
Posts [lost / found]  비고 : 게시물 종 류로 URL 구분, found 경우 {storage} 파라미터 추가	POST	/api/lost/create /api/found/create		Data : { title : String, category : String, date : datetime, area : String, place : String, item : String, contents : String, state : String, {storage : String} }, Image : MultipartFile[]	200 작성 완료 : String	게시물 작성	USER
	GET	/api/lost/read/{postId}	postId : Long		200	해당 게시물	

		/api/found/read/{postId}			PostInfo : { postId : Long title : String, images : List<String>, category : String, date : datetime, area : String, place : String, item : String, contents : String, state : String, traderTag : String time : datetime {storage : String} }, PostUser : { nickname : String, profile : String, tag : String } 404	상세 조회	
	GET	/api/lost/read /api/found/read		page : int(0), Size : int(20)	200 Page<List<>> { postId : Long, title : String, image : String, area : String, time : datetime }	전체 게시물 목록 조회	

	GET	/api/lost/read/user/{tag} /api/found/read/user/{tag}	tag : String		200 List<> : {: postId : Long, title : String, images : List<String>, category : String, date : datetime, area : String, place : String, item : String, contents : String, state : String, traderTag : String, time : datetime, {storage : String} }	선택한 사용자가 작성한 게시물 목록 조회	
	GET	/api/lost/search /api/found/search		{ title : String, category : String, date : datetime, area : String, place : String, item : String, contents : String, state : String, page : int(0), size : int(20) }	200 Page<List<>> : { postId : Long, title : String, image : String, area : String, time : datetime }	검색 필터 적용 후 조회	
	GET	/api/lost/read/chat/{postId} /api/found/read/chat/{postId}	postId : Long		200 List <> {	채팅자 태그 조회	USER

					username : String, tag : String }		
	PATCH	/api/lost/update /api/found/update		Data : { postId : Long, title : String, category : String, date : datetime, area : String, place : String, item : String, contents : String, state : String, images : String {storage : String} }, Images : MultipartFile[]	200 수정 완료 : String	게시물 수정	USER
	PATCH	/api/lost/change /api/found/change		{ postId : Long state : String, traderTag : String }	200 변경 완료 : String	게시물 상태 수정	USER
	DELETE	/api/lost/delete /api/found/delete		{ postId : Long }	200 삭제 완료 : String	게시물 삭제	USER
Keyword	POST	/api/keyword/create		{ Keyword : String }	200 등록 완료 : String	키워드 생성	USER

	GET	/api/keyword/read			200 List<> { keyword : String, time : datetime }	본인 키워드 조회	USER
	GET	/api/keyword/search			200 List<> { postId : Long, type : String, title : String, image : String, area : String, time : datetime }	키워드 검색 결과 조회	USER
	PATCH	/api/keyword/update		{ beforeKeyword : String, afterKeyword : String }	200 수정 완료 : String	키워드 수정	USER
	DELETE	/api/keyword/delete		{ keyword : String }	200 삭제 완료 : String	키워드 삭제	USER
Report	POST	/api/reports/create		{ userTag : String, type : String, location : Long title : String, contents : String }	200 완료 : String	신고 글 작성 (신고하기)	USER
Review	POST	/api/review/create		{ star : float, }	200 완료 : String	평가글 작성	USER

				postType : String, postId : Long, contents : String }	201 이미 평가글을 작성 : String 202 작성 권한 없음 : String		
	GET	/api/review/read/{tag}	tag : String		200 List<> : { reviewedUserTag : String, reviewedNickname : String, role : String, contents : String time : datetime }	평가글 조회	
	DELETE	/api/review/delete		{ reviewId : Long }	200 삭제 완료 : String	평가글 삭제	USER
Chat	POST	/api/chat/room/create		{ postId : Long, postType : String, message : String }	200	채팅방 개설	USER
	GET	/api/chat/room/read/user			200 List<> { roomId : Long, { profile : String, nickname : String, tag : String }, { lastMessage : String,	본인의 채팅방 목록 조회	USER



					time : datetime }		
	GET	/api/chat/room/read/{roomId}	roomId : Long		200 { { roomId : Long, hostUserTag : String, guestUserTag : String, leaveUserTag : List<String>, }, { postId : Long, postImg : String, title : String, state : String, traderTag : String }, { profile : String, nickname : String, tag : String } }	특정 채팅방 정보 조회	USER
	GET	/api/chat/get/list		{ postId : Long, postType : String }	200 List<> : { roomId ;; Long, { profile : String, nickname : String, tag : String }, { lastMessage : String,	본인 게시물 상세 페이지에서 채팅방 목록 조회	USER

					time : datetime } }		
	GET	/api/chat/get/post		{ postId : Long, postType : String } 	200 { roomId : Long } 	게시물에 생성된 본인의 채팅방 조회	USER
	GET	/api/chat/get/room/{roomId}	roomId : Long		200 List<> : { messageId : Long message : String, sender : String, time : datetime } 	채팅방의 전체 메시지 조회	USER
	DELETE	/api/chat/room/leave		{ roomId : Long } 	200	채팅방 나가기	USER