대분류	추가능	METHOD	URL	BODY 백에서 알아야할 정	RETURN	http 상태 코드	설명

유저	회원가입	POST	/api/user	RequestUserDto
				{
				email: "gildong@gmail.com",
				password: "1234",
				nickname: "gildongnickname",
				profileImg: "2"
				}
					{
					email: "gildong@gmail.com",
					nickname: "gildongnickname",
					profileImg: 2,
					classic_pt: 0,
					item_pt: 0,
					winNum: 0,
					loseNum: 0
					messages: []
					}
							회원가입(추후 수정) userId는 토큰으로 인식 가능

	로그인	POST	/api/auth/login	RequestAuthDto
				{
				email: "gildong@gmail.com",
				password: "1234"
				}
					{
					email: "gildong@gmail.com,
					nickname: "gildongnickname",
					profileImg: 2,
					classic_pt: 50,
					item_pt: 235,
					winNum: 3,
					loseNum: 5,
					messages: [{
					userFrom: "youngheo",
					userTo: "cheolsu",
					isRead: false,
					msgText: "안녕하세요",
					msgType: 1
					}]
					}		

	로그아웃	GET	/api/auth/logout		status: 200		

	정보수정	PATCH	/api/user	RequestModifyUserDto
				{
				currentpassword: "currentpassword",
				newpassword: "newpassword",
				nickname: "newnickname",
				profileImg: "3"
				}
					X		수정할 격체만 파라미터에 담을 것

	닉네임 중복확인	GET	/api/check/nickname?=nickname"		{
					check: true
					}		

	이메일 중복확인	GET	/api/check/email?=email"		{
					check: true
					}		

	비밀번호 확인	POST	/api/check/password	RequestUserDto
				{
				password:"1234"
				}
					비밀번호 맞지 시 : 200
					비밀번호 불맞지 시 : 401		

	회원탈퇴	DELETE	/api/user				

	유저 검색	GET	/api/user/{searchWord}		{
					email: "gildong@gmail.com,
					nickname: "gildongnickname",
					profileImg: 2,
					classic_pt: 50,
					item_pt: 235,
					winNum: 5,
					loseNum: 3
					}		

친구 관리	친구 목록 확인	GET	/api/friend		[
					{
					email: "friend@gmail.com", nickname: "friend", profileImg: 3,
					classic_pt: 30, item_pt : 360, winNum: 5, loseNum: 3
					},
					{
					email: "friend2@gmail.com", nickname: "friend2", profileImg: 3,
					classic_pt: 30, item_pt : 360, winNum: 6, loseNum: 4
					}
					]		

	친구 요청	POST	/api/friend	RequestFriendDto
				{
				userTo : "userId"
				}
							친구 관계 추가 후, 친구 요청 메세지 삭제

	친구 삭제	DELETE	/api/friend/{userId}		status: 200		
	친구 따라가기	모름					
랭크 게임	매칭 요청	GET	/api/rank/matching	Token			webSocket을 통해 openvidu 방 토큰, 매칭된 유저 정보 같이 return

	랭크 게임이 끝나면 게임 결과를 테이블에 저장	POST	/api/rank/result	RequestRankDto
				{
				is_item : true
				user_winner: 2
				user_loser : 3
				}
					status 200		

	랭크 게임이 끝나면 얻은 포인트를 유저에 반영	PATCH	/api/rank/{userId}	RequestPointrDto
				{
				classicPt : 123 or itemPt : -123
				}
					status 200		

일반 게임	게임 방 만들기	POST	/api/room	RequestCreateRoomDto
				{
				roomName: "gildongRoom",
				password: "1234" or "",
				ownerId: "3",
				capacity: "3",
				isItem: "true"
				}			

	게임 방 목록 조회	GET	/api/room/Item		방 목록(페이지네이션)		
		GET	/api/room/classic		방 목록(페이지네이션)		

	게임 방 들어가기 (토큰 요구하기)	POST	/api/room/enter	{
				roomName: "gildongRoom",
				password: "1234", ""
				}
					openvidu token,
					예외: 비밀번호가 틀림, 방이 꽉 참, 블랙리스트 유저임, 방이 없어진 상태임		

	빠른 참가	GET	/api/room/quick		openvidu token, 예외: 참가할 수 있는 방이 없음		

	게임 방 나가기	POST	/api/room	{
				roomId: "id"
				}			

강퇴 하기(블랙리스트 추가		POST	/api/blackList	RequestBlackListDto
				{
				userId : "id",
				roomId : "id"
				}			

메세지	메세지 리스트 확인	GET	/api/message		[{msgType: 1, msgText: "안녕하세요", isRead: false, userFrom: "gildong", createdAt: "2023-12-21-09-12"}]		

	메세지 확인	PATCH	/api/message/{messageId}	{
				isRead: true
				}			

	메세지 보내기	POST	/api/message	RequestMessageDto
				{
				msgType : 0,
				msg_text : "Hi",
				user_to : "to",
				userFrom : "from"
				}			

	친구 추가 소락	POST	/api/message/friend	RequestFriendDto
				{
				userFrom : "userId"
				}			

	시스템 메세지 수락/거절 후 메세지 삭제	DELETE	/api/message/{messageId}				친구 거절 시 메세지 삭제
랭킹 조회	클래식 랭킹 조회	GET	/api/rank/classic		[{nickname: "gildongnickname", profileImg: 2, classic_pt : 1000, winNum : 10, loseNum: 20}]		
	아이템전 랭킹 조회	GET	/api/rank/item		[{nickname: "gildongnickname", profileImg: 2, item_pt : 1000, winNum : 10, loseNum: 20}]		
Controller	userController						
	authController						
	friendController						
	recordController						
	rankingController						
	roomController						
	messageController						
	blacklistController						
	openViduController						