

Communication Protocol

2021.10.4 최신화

□ Matching Table(Android <-> Server(Spring))

ID	Microservice	url	Method	Request	Response	비고
1	user-service	/sendRegistCertification	POST	Header : No Body : RequestUser(phoneNum, hashKey)	Success : 200	# 회원가입(핸드폰 인증을 위한 서버에 문자 전송 요청) 사용자는 회원가입시 휴대폰 인증을 한다.
					Fail : 400 Body : 실패 메시지	사용자는 핸드폰 인증을 위해서 핸드폰 번호를 입력한다. 사용자는 인증번호를 문자로 받는다. 사용자는 받은 인증번호를 어플리케이션에 입력한다.
2	user-service	/checkRegistCertification	POST	Header : No Body : RequestUser(phoneNum, numberId)	Success : 200	# 회원가입(핸드폰 인증을 위한 서버에 인증번호 전송) 사용자가 입력한 인증번호와 DB의 인증번호 가 같은지 확인한다.
					Fail : 400 Body : 실패 메시지	사용자가 입력한 인증번호와 DB에 저장한 인증번호가 같다면 인증한 시간을 DB에 저장한다.
3	user-service	/regist	POST	Header : No Body : RequestUser(phoneNum, email, password, nickname, birthdate, gender, other_M, other_W, respect, contact, date, communication, sex, work, happy, dating, profileImage, profileImageIc on, locationKOR, latitude, longitude, sublocationKOR, subLatitude, subLongitude, userGrade, minAge, maxAge, maxDistance)	Success : 200 header : userToken	# 회원가입(서버에 정보 전송) 사용자는 사용자의 위치정보를 입력한다(googleMap이용, pointer) 사용자는 email, password checkpasseord를 입력한다. 사용자는 nickName, birth,gender,matchGender를 입력 및 선택한다. (nickname,birth 는 입력 gender는 선택 matchGender는 중복선택가능) 사용자는 moodDistance탐색 질문(respect, context, date, communication, sex 순)을 선정한다. 사용자는 연애관 질문 (work, happy, dating)에 관한 것들을 입력한다. 사용자는 profileImage를 선정한다.
					Fail : 400	
4	user-service	/checkEmail	POST	Header : No Body : RequestUser(email)	Success : 200	# 회원가입(이메일에 대한 중복 확인 : 같은 메일은 1개만 허용) 사용자는 회원가입시 email을 입력한다. 이때 email이 중복된 email인지 확인한다.
					Fail : 400 Body : 실패 메세지 - ResponseUser(userUid): "overlap" - ResponseUser(userUid): "error"	
5	user-service	/sendCertification	POST	Header : No Body : RequestUser(phoneNum, hashKey)	Success : 200	# ID / PW 찾기(핸드폰 인증을 위한 서버에 문자 전송 요청) 사용자의 핸드폰으로 인증번호를 전송(hashKey포함)한다. Databases 상에 같은 전화번호가 있으면, 둘다 enable처리한다. password 재설정에서 보내지는 email에 대한 password 변경처리
					Fail : 400 Body : 실패 메세지 - ResponseUser(userUid): "overlap" - ResponseUser(userUid): "error"	
6	user-service	/certificateNumber	POST	Header : No Body : RequestUser(phoneNum, numberId)	Success : 200	# ID / PW 찾기(핸드폰 인증을 위한 서버에 문자 전송 요청) 사용자가 입력한 인증번호가 같은 지 확인 올바르지 않은 값일 경우, 실패
					Fail : 400 Body : 실패 메세지 - ResponseUser(userUid): "wrong" - ResponseUser(userUid): "error"	
7	user-service	/findByEmail	POST	Header : No Body : RequestUser(phoneNum)	Success : 200 Body : ResponseUser(email)	# ID / PW 찾기(ID 찾기 : 핸드폰 번호로 Email을 찾아준다.) 사용자가 핸드폰 번호를 입력한다.
					Fail : 400 Body : 실패메세지 - ResponseUser(userUid): "no data" - ResponseUser(userUid): "error"	사용자의 Email을 조회 후 반환한다.
8	user-service	/findByPassword	POST	Header : No Body : RequestUser(phoneNum,email)	Success : 200 header : userToken	# ID / PW 찾기(PW 찾기 : 핸드폰 번호와 Email에 해당하는 사용자토큰을 생성해준다.)
					Fail : 400 Body : 실패메세지 - ResponseUser(userUid): "no data" - ResponseUser(userUid): "error"	사용자 Password 재설정을 위해 토큰을 보내준다.
9	user-service	/resetPassword	POST	Header : userToken Body : ResqusetUser(email, phoneNum, password)	Success : 200	# ID / PW 찾기(PW 재설정 : 사용자토큰, 핸드폰 번호, Email과 일치하는 사용자를 찾아, PW를 변경한다.)
					Fail : 400	사용자가 이메일을 입력한다. 사용자가 재설정하려는 Password를 입력한다. 사용자가 입력한 해당 이메일에 해당하는 패스워드를 변경해준다.
10	user-service	/login	POST	Header : No Body : ResquestUser(email, password)	Success : 200 header : userToken Body : ResponseUser(All user data of VO)	# 로그인 사용자가 로그인(email, password)을 시도한다 사용자가 입력한 email과 password가 맞으면 사용자 정보를 반환해준다.
					Fail : 400	
11	user-service	/autologin	POST	Header : userToken	Success : 200 header : userToken Body : ResponseUser(All user data of VO)	# 자동 로그인 사용자가 토큰을 보유할 시에 자동 로그인을 해준다/ 사용자가 로그인에 성공하면 갱신 된 사용자 Token을 반환해 준다.
					Fail : 400	
12	post-service	/getPosts/{page}/{postType}	POST	Header : userToken	Success : 200 Body : List(ResponsePost(Post_id, profileImageIc on, userUid, nickname, locationKOR, locationENG, postLikeCount, postCommentCount, postTime, postImage, postContents, page))	# 게시물 서비스(게시물 가져오기) 게시물 가져오기 - postType : □ 전체 : String(total) □ 인기 : String(popular) □ 메인 지역 : String(areaA) - page : 페이지 번호(페이지 당 100개씩 가져옴) - profileImageIc on : 포스트를 게시한 사람의 프로필 아이콘 - userUid : 포스트를 게시한 사람의 고유식별자 - nickname : 포스트를 게시한 사람의 닉네임
					Fail : 400	
13	post-service	/clickLike/{postId}/{commentId}	POST	Header : userToken Body : RequestPost(postId, liked(commentId))	Success : 200 Body : Boolean(true or false)	# 게시물 서비스(좋아요 기능) postId인 게시물의 commentId인 댓글(게시물에 대한 좋아요라면 NULL) 해당 좋아요가 눌리면 true, 취소되면 false
					Fail : 400	
14	post-service	/registComment	POST	Header : userToken Body : RequestComment(postId, commentContents, commentClass, commentGroup)	Success: 200	# 게시물 서비스(댓글, 대댓글 등록) postId : 게시물의 고유식별자 commentContents : 댓글의 내용 commentClass : 댓글(0) or 대댓글(1) commentGroup : 댓글의 그룹 ex 1) 새로 생성하는 댓글 commentClass(0), commentGroup(null) : commentGroup은 서버에서 찾아 서 등록 ex 2) 새로 생성하는 대댓글 commentClass(1), commentGroup(해당 댓글의 Group) ex 3) 기존에 댓글이 있는 곳에 등록하는 댓글(commentClass(0),
					Fail : 400	

						commentGroup(null) ex 4) 기존에 대댓글이 있는 곳에 등록하는 댓글 commentClass(1), commentGroup(해당 댓글의 Group)
15	user-service	/getUser	POST	Header : userToken Body : RequestProfile(userUid)	Success : 200 Body: RequesetProfile(profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, work, happy, dating, moodDistance, respect, contact, date, communication, sex) Fail : 400	# 프로필 조회(대상 프로필 가져오기) #Mood(프로필 가져오기) 기능 및 본인, 상대 프로필 가져오기
16	post-service	/getPosts/{page}	POST	Header : userToken Body : RequesetPost(postUid)	Success : 200 Body : List(ResponsePost(post_id, profileImagelcon, userUid, nickname, locationKOR, locationENG, postLikeCount, postCommentCount, postTime, postImage, postContents)) Fail : 400	# 프로필 조회(대상 게시물 가져오기) 보낸 사용자(Uid)의 게시물 리스트가 보여주는 기능(본인 or 상대 게시물)
17	post-service	/registPost	POST	Header : userToken Body : ResquestPost(locationKOR, locationENG, postImage, postContents, List(hashtag(hashtagName)))	Success : 200 Fail : 400	# 게시물 서비스(게시물 등록) 게시물 작성
18	post-service	/updatePost	POST	Header : userToken Body : RequesetPost(postId, postContents, List(hashtag(hashtagName)))	Success : 200 Fail : 400	# 게시물 서비스(작성된 게시물 수정) 게시물 수정(사진 수정 불가)
19	post-service	/deletePost	POST	Header : userToken Body : RequestPost(postId)	Success : 200 Fail : 200	# 게시물 서비스(작성된 게시물 삭제) 게시물 삭제
20	matching-service	/getMatchingUsers	POST	Header : userToken	Success : 200 Body : List(ResponseMatchingUser(userUid, profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, respect, contact, date, communication, sex MatchingData(moodDistance, matchingTime))) Fail : 400	# 매칭 서비스(매칭된 상대 정보 가져오기) 매칭상대정보(2명) 가져오기
21	matching-service	/updateMatchingTime	POST	Header : userToken String("matching or profile or request or reset")	Success : 200 Body : List(ResponseMatchingUser(userUid, profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, respect, contact, date, communication, sex MatchingData(moodDistance, matchingTime))) Fail : 400	# 매칭 서비스(광고 시청 후 시간 단축시키기) 광고 시청으로 시간 단축 - 매칭 페이지 광고 : String("matching") - 프로필 페이지 광고 : String("profile") - 받은 요청 페이지 광고 : String("request") - 탈퇴 시도 시, 1회 매칭 리셋(6개월에 1번 리셋) : String("reset")
22	user-service	/purchaseVIP	POST	Header : userToken RequestPurchase(orderId, packageName, productId, purchaseTime, purchaseState, purchaseToken, acknowledged)	Success : 200 Body : ResponseUser(All user data of VO) Fail : 400	# 결제 서비스(결제에 대한 서비스 제공 요청) 유료 결제 시, 결제한 시간에 한번, 하루에 한번 지급(0시 기준). 부여되는 코인의 갯수는 테이블에서 가져와서 조정가능하게 함 - purchaseContents : monthly_item_1(1개월 권), monthly_item_2(6개월 권), monthly_item_3(12개월 권)
23	user-service	/setMatching	POST	Header : userToken RequesetUser(All user data of VO)	Success : 200 Fail : 400	# 사용자 설정(사용자의 설정 변경) 매칭설정(상대성별 설정, 최대거리 설정, 상대 연령대 설정, 닉네임설정(코인), 무드거리설정(코인))(코인or티켓 설정), 프로필, 프로필아이콘 유료 기능 : 핸드폰, 닉네임, 무드거리, 로밍, 지역
24	user-service	/blockPhoneNums	POST	Header : userToken Body : List(phoneNum)	Success : 200 Fail : 400	# 사용자 설정(연락처 차단 요청) 연락처 차단(사용자가 지정한 번호 목록 차단)
25	user-service	/updatePhoneNum	POST	Header : userToken Body : RequestUser(phoneNum)	Success : 200 Fail : 400	# 사용자 설정(핸드폰 번호 변경) 전화번호 재설정 : (유료기능)핸드폰 인증 성공하면(6번) 받은 핸드폰 번호로 수정하여 저장
26	user-service	/openCheating	POST	Header : userToken Body : Integer(coin), Integer(ticket)	Success : 200 Fail : 400	# 채팅 서비스(채팅방 개설로 인한 요금 차감) 채팅방 개설(유료기능)
27	post-service	/getHashtag/{hashtagName}/{page}	POST	Header : userToken	Success : 200 Body : List(ResponsePost(post_id, profileImagelcon, userUid, nickname, locationKOR, locationENG, postLikeCount, postCommentCount, postTime, postImage, postContents)) Fail : 400	# 게시물 서비스(해쉬태그로 게시물 검색) 해쉬태그로 검색한 게시물들에 대한 리스트
28	lock-service	/addLockUser	POST	Header:userToken Body : RequestLockUser(lockUserUid, lockType, lockReasons, referUid)	Success : 200 Fail : 400	# 신고 서비스(대상 신고) 신고하기 기능(제재 대상, 제재타입, 제재사유, 참고할 값의 UID) 제재 타입 : 프로필사진 신고("profileImage"), 프로필 질문 신고("profileQuestion"), 프로필닉네임 신고("profileNickname"), 채팅신고("cheating")
29	post-service	/getPost/{postId}	GET	Body : String(postId)	Success : 200 Body: ResponsePost(All data of VO) Fail : 400	#게시물 서비스(해당 게시물 가져오기) 추후 SNS서비스에 URL공유만으로 게시물 공유가 가능하게끔 GET으로 구현.
30	user-service	/checkNickname	POST	Header : No Body : RequestUser(nickname)	Success : 200 Fail : 400	# 회원가입(닉네임 중복 확인)
31	user-service	/getBlockPhoneNums	POST	Header : userToken Body : No	Success : 200 Body : List<String> phoneNum Fail : 400 : null	차단된 전화번호 목록
32	post-service	/deleteComment	POST	Header : userToken	Success : 200	댓글 삭제

				Body: RequestComment(commentId)	Fail : 400	
--	--	--	--	---------------------------------	------------	--

□ VO(Android <-> Server(Spring))

ID	Microservice	VO(Only Communication Model)	Parameter	Parameter 설명
1	user-service	RequestUser	<div><div>-</div><div>userId</div><div>-</div><div>phoneNum</div><div>-</div><div>email</div><div>-</div><div>password</div><div>-</div><div>nickname</div><div>-</div><div>birthdate</div><div>-</div><div>gender</div><div>-</div><div>otherM</div><div>-</div><div>otherW</div><div>-</div><div>respect</div><div>-</div><div>contact</div><div>-</div><div>date</div><div>-</div><div>communication</div><div>-</div><div>sex</div><div>-</div><div>work</div><div>-</div><div>happy</div><div>-</div><div>dating</div><div>-</div><div>profileImage</div><div>-</div><div>profileImagelcon</div><div>-</div><div>locationKOR</div><div>-</div><div>latitude</div><div>-</div><div>longitude</div><div>-</div><div>sublocationKOR</div><div>-</div><div>subLatitude</div><div>-</div><div>subLongitude</div><div>-</div><div>userGrade</div><div>-</div><div>minAge</div><div>-</div><div>maxAge</div><div>-</div><div>maxDistance</div><div>-</div><div>numberId</div><div>-</div><div>hashkey</div></div>	<div><div>-</div><div>사용자의 고유식별자</div><div>-</div><div>핸드폰 번호</div><div>-</div><div>이메일</div><div>-</div><div>비밀번호</div><div>-</div><div>별명</div><div>-</div><div>생년월일</div><div>-</div><div>성별</div><div>-</div><div>매칭하고자 하는 상대(남), 희망하면 True, 비희망하면 False</div><div>-</div><div>매칭하고자 하는 상대(여), 희망하면 True, 비희망하면 False</div><div>-</div><div>무드거리 판별질문 1 : 존중 (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 2 : 연락 (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 3 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 4 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 5 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>질문 1 : 어떤 일을 하는지?</div><div>-</div><div>질문 2 : 얼마나 행복한지?</div><div>-</div><div>질문 3 : 얼마나 데이트를 중요하게 여기는지?</div><div>-</div><div>프로필 이미지</div><div>-</div><div>프로필 아이콘</div><div>-</div><div>지역 위치(한글)</div><div>-</div><div>위도</div><div>-</div><div>경도</div><div>-</div><div>(로밍을 위한) 지역 위치(한글)</div><div>-</div><div>(로밍을 위한) 위도</div><div>-</div><div>(로밍을 위한) 경도</div><div>-</div><div>사용자 등급(VIP : VIP, Other : Basic</div><div>-</div><div>최소 나이(Default : ((생년월일로 계산된 나이) - 5 >=20)</div><div>-</div><div>최대 나이(Default : ((생년월일로 계산된 나이) +5))</div><div>-</div><div>최대 거리(Default : 50km)</div><div>-</div><div>인증번호(핸드폰 인증시, 사용)</div><div>-</div><div>핸드폰 문자 전송 시, 같이 보내줄 해쉬키</div></div>
2	user-service	ResponseUser	<div><div>-</div><div>userToken</div><div>-</div><div>userId</div><div>-</div><div>phoneNum</div><div>-</div><div>email</div><div>-</div><div>nickname</div><div>-</div><div>birthdate</div><div>-</div><div>gender</div><div>-</div><div>otherM</div><div>-</div><div>otherW</div><div>-</div><div>respect</div><div>-</div><div>contact</div><div>-</div><div>date</div><div>-</div><div>communication</div><div>-</div><div>sex</div><div>-</div><div>work</div><div>-</div><div>happy</div><div>-</div><div>dating</div><div>-</div><div>profileImage</div><div>-</div><div>profileImagelcon</div><div>-</div><div>locationKOR</div><div>-</div><div>locationENG</div><div>-</div><div>latitude</div><div>-</div><div>longitude</div><div>-</div><div>subLocationKOR</div><div>-</div><div>subLocationENG</div><div>-</div><div>subLatitude</div><div>-</div><div>subLongitude</div><div>-</div><div>userGrade</div><div>-</div><div>minAge</div><div>-</div><div>maxAge</div><div>-</div><div>maxDistance</div><div>-</div><div>ticket</div><div>-</div><div>coin</div><div>-</div><div>resetMatching</div><div>-</div><div>userLock</div><div>-</div><div>userLockReasons</div></div>	<div><div>-</div><div>사용자의 Token(로그인 시, 발급. 로그아웃 시, 폐기)</div><div>-</div><div>사용자의 고유식별자</div><div>-</div><div>핸드폰 번호</div><div>-</div><div>이메일</div><div>-</div><div>별명</div><div>-</div><div>생년월일</div><div>-</div><div>성별</div><div>-</div><div>매칭하고자 하는 상대(남), 희망하면 True, 비희망하면 False</div><div>-</div><div>매칭하고자 하는 상대(여), 희망하면 True, 비희망하면 False</div><div>-</div><div>무드거리 판별질문 1 : 존중 (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 2 : 연락 (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 3 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 4 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>무드거리 판별질문 5 : (수정 시, 티켓 or 코인 필요)</div><div>-</div><div>질문 1 : 어떤 일을 하는지?</div><div>-</div><div>질문 2 : 얼마나 행복한지?</div><div>-</div><div>질문 3 : 얼마나 데이트를 중요하게 여기는지?</div><div>-</div><div>프로필 이미지</div><div>-</div><div>프로필 아이콘</div><div>-</div><div>지역 위치(한글)</div><div>-</div><div>지역 위치(영어)</div><div>-</div><div>위도</div><div>-</div><div>경도</div><div>-</div><div>(로밍을 위한) 지역 위치(한글)</div><div>-</div><div>(로밍을 위한) 지역 위치(영어)</div><div>-</div><div>(로밍을 위한) 위도</div><div>-</div><div>(로밍을 위한) 경도</div><div>-</div><div>사용자 등급(VIP : VIP, Other : normal</div><div>-</div><div>최소 나이(Default : ((생년월일로 계산된 나이) - 5 >=20)</div><div>-</div><div>최대 나이(Default : ((생년월일로 계산된 나이) +5))</div><div>-</div><div>최대 거리(Default : 50km)</div><div>-</div><div>티켓(유료 자산)</div><div>-</div><div>코인(유료 자산)</div><div>-</div><div>사용자가 탈퇴 시도 시, 매칭 다시 시도(6개월 1회)B</div><div>-</div><div>사용자 잠금 기능(boolean(default : false))</div><div>-</div><div>제재사유 : default(null) profile신고(“profile”), 게시물신고(“post”), 채팅신고(“cheating”), 댓글신고 (“comment“)</div></div>
3	post-service	RequestPost	<div><div>-</div><div>postId</div><div>-</div><div>postUid</div><div>-</div><div>locationENG</div><div>-</div><div>locationKOR</div><div>-</div><div>postImage</div><div>-</div><div>postContents</div><div>-</div><div>commentContents</div><div>-</div><div>commentClass</div><div>-</div><div>commentGroup</div><div>-</div><div>List: hashtag<div><div>□</div><div>hashtagName</div></div><div>-</div><div>List: liked<div><div>□</div><div>commentId</div></div></div></div></div>	<div><div>-</div><div>게시물에 대한 고유 식별자</div><div>-</div><div>게시물을 올린 사용자의 고유식별자</div><div>-</div><div>지역(영어)</div><div>-</div><div>지역(한글)</div><div>-</div><div>게시물 사진</div><div>-</div><div>게시물 내용</div><div>-</div><div>대, 댓글 내용</div><div>-</div><div>댓글 분류</div><div>-</div><div>댓글 그룹</div><div>-</div><div>해쉬태크에 대한 리스트<div><div>□</div><div>해쉬태그 이름</div></div><div>-</div><div>like 객체 리스트<div><div>□</div><div>댓글에 대한 고유 식별자</div></div></div></div></div>
4	post-service	ResponsePost	<div><div>-</div><div>postId</div><div>-</div><div>profileImagelcon</div><div>-</div><div>postUid</div></div>	<div><div>-</div><div>게시물의 고유 식별자(게시물을 추적가능하게 하는 역할)</div><div>-</div><div>포스트를 게시한 사람의 프로필 아이콘</div><div>-</div><div>포스트를 게시한 사람의 고유 식별자</div></div>

-

사용자의 고유식별자

-

핸드폰 번호

-

이메일

-

비밀번호

-

별명

-

생년월일

-

성별

-

매칭하고자 하는 상대(남), 희망하면 True, 비희망하면 False

-

매칭하고자 하는 상대(여), 희망하면 True, 비희망하면 False

-

무드거리 판별질문 1 : 존중 (수정 시, 티켓 or 코인 필요)

-

무드거리 판별질문 2 : 연락 (수정 시, 티켓 or 코인 필요)

-

무드거리 판별질문 3 : (수정 시, 티켓 or 코인 필요)

-

무드거리 판별질문 4 : (수정 시, 티켓 or 코인 필요)

-

무드거리 판별질문 5 : (수정 시, 티켓 or 코인 필요)

-

질문 1 : 어떤 일을 하는지?

-

질문 2 : 얼마나 행복한지?

-

질문 3 : 얼마나 데이트를 중요하게 여기는지?

-

프로필 이미지

-

프로필 아이콘

-

지역 위치(한글)

-

위도

-

경도

-

(로밍을 위한) 지역 위치(한글)

-

(로밍을 위한) 위도

-

(로밍을 위한) 경도

-

사용자 등급(VIP : VIP, Other : Basic

-

최소 나이(Default : ((생년월일로 계산된 나이) - 5 >=20)

-

최대 나이(Default : ((생년월일로 계산된 나이) +5))

-

최대 거리(Default : 50km)

-

인증번호(핸드폰 인증시, 사용)

-

핸드폰 문자 전송 시, 같이 보내줄 해쉬키

			<ul style="list-style-type: none"> - nickname - locationKOR - locationENG - postLikeCount - postCommentCount - postTime - postImage - postContents - List: commentInfo <ul style="list-style-type: none"> □ commentId □ commentUid □ nickname □ profileImageIcon □ commentContents □ commentClass □ commentGroup □ commentTime □ commentLikeCount 	<ul style="list-style-type: none"> - 포스트를 게시한 사람의 별명 - 포스트를 게시한 사람의 지역(한글) - 포스트를 게시한 사람의 지역(영어) - 포스트의 좋아요 갯수 - 포스트의 댓글 갯수 - 포스트를 올린 시간 - 포스트의 사진 - 포스트의 내용(Hashtag가 내용안에 포함되어 있어, 따로 넣어주지 않아도 됨) - 댓글, 대댓글에 대한 사항 <ul style="list-style-type: none"> □ 댓글의 ID(고유식별자) □ 댓글을 달은 사용자의 고유식별자 □ 댓글을 달은 사용자의 닉네임 □ 댓글을 달은 사용자의 프로필 아이콘 □ 댓글의 내용 □ 댓글인지, 대댓글인지 구분(0:댓글, 1:대댓글) □ 댓글에 대한 그룹핑(같은 숫자이면 같은 댓글안에 포함) □ 댓글을 달은 시간(서버시간 기준) □ 댓글의 좋아요 수
5	user-service	RequesetProfile	<ul style="list-style-type: none"> - userId 	<ul style="list-style-type: none"> - 요청된 사용자의 고유식별자
6	user-service	ResponseProfile	<ul style="list-style-type: none"> - profileImage - profileIcon - nickname - birthdate - location - physicalDistance - work - happy - dating - moodDistance - respect - contact - date - communication - sex 	<ul style="list-style-type: none"> - 요청된 프로필의 사진 - 요청된 프로필의 아이콘 - 요청된 사용자의 별명 - 요청된 사용자의 생년월일 - 요청된 사용자의 지역(영어) - 요청된 사용자의 물리적 거리 - 질문 1 : 어떤 일을 하는지? - 질문 2 : 얼마나 행복한지? - 질문 3 : 얼마나 데이트를 중요하게 여기는지? - 나와의 Mood거리 - 무드거리 판별질문 1 : 존중 (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 2 : 연락 (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 3 : (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 4 : (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 5 : (수정 시, 티켓 or 코인 필요)
7	matching-service	ResponseMatchingUser	<ul style="list-style-type: none"> - userId - profileImage - profileIcon - nickname - birthdate - locationKOR - locationENG - physicalDistance - respect - contact - date - communication - sex - MatchingData <ul style="list-style-type: none"> □ moodDistance □ matchingTime 	<ul style="list-style-type: none"> - 매칭상대의 고유식별자 - 프로필 사진 - 프로필 아이콘 - 매칭상대의 별명 - 매칭상대의 생년월일 - 지역(한글) - 지역(영어) - 물리적 거리 - 무드거리 판별질문 1 : 존중 (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 2 : 연락 (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 3 : (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 4 : (수정 시, 티켓 or 코인 필요) - 무드거리 판별질문 5 : (수정 시, 티켓 or 코인 필요) - 매칭한 데이터 <ul style="list-style-type: none"> □ 무드 거리 □ 매칭 완료 시간
8	user-service	RequestPurchase	<ul style="list-style-type: none"> - orderId - packageName - productid - purchaseTime - purchaseState - purchaseToken - acknowledged 	<ul style="list-style-type: none"> - 구매 식별자 - 구매 패키지 이름 - 구매 제품 식별자 - 구매한 시간 - 구매한 상태 - 구매 토큰 - 인정여부
9	lock-service	RequestLockUser	<ul style="list-style-type: none"> - lockUserId - lockType - lockReasons - referUid 	<ul style="list-style-type: none"> - 신고대상 UID - 신고타입 : 프로필사진 신고("profileImage"), 프로필 질문 신고("profileQuestion"), 프로필닉네임 신고("profileNickname"), 채팅신고("cheating") - 신고 사유 - 참조값 UID
10	user-service	RequestBlockUser	<ul style="list-style-type: none"> - phoneNum - disabled 	<ul style="list-style-type: none"> - 차단할 핸드폰 번호 - 차단 여부
11	user-service	ResponseBlockUser	<ul style="list-style-type: none"> - phoneNum - disabled 	<ul style="list-style-type: none"> - 차단할 핸드폰 번호 - 차단 여부
12	post-service	RequestComment	<ul style="list-style-type: none"> - postId - commentId - commentContents - commentClass - commentGroup 	<ul style="list-style-type: none"> - 게시물 식별자 - 댓글 식별자 - 댓글 내용 - 댓글 분류 - 댓글, 대댓글 그룹(0이면 댓글, 1이면 대댓글)

□ JSON Value

ID	Microservice	url	Method	Request	Response
1	user-service	/sendRegistCertification	POST	Header : No Body : RequestUser(phoneNum, hashKey)	m Fail : 400 Body : 실패 메시지
2	user-service	/checkRegistCertification	POST	Header : No Body : RequestUser(phoneNum, numberId)	Success : 200 Fail : 400 Body : 실패 메시지

3	user-service	/regist	POST	<div>Header : No</div> <div>Body :<pre>{ "phoneNum": "01012341234", "email": "seojeonghyeon@gmail.com", "password": "test123456", "nickname": "Lime-IT", "birthdate": "1995-01-01", "gender": true, "otherM": false, "otherW": true, "respect": "5", "contact": "3", "date": "5", "communication": "4", "sex": "3", "work": "그들은 이상은 고행을 곳이 들어 황금시대다. 새가 밥을 이 되려 니와, 전환하였는가? 사랑의 인생에 수 그리하였는가? 피고, 사랑의 장식하는 우리 때문이다. 같이, 되는 이것이야말로 불러 풍부하게 고통을 보내는 우리는 아니다. 찬미를 동력은 같지 뿐이다. 수 것이 주며, 인류의 자신과 이상의 충분 히 열락의 아니다. 가치를 더운지라 석가는 생명을 같은 풀이 것이다. 보라, 없 으면 같지 것이다. 예수는 이상을 있는 힘있다. 따뜻한 커다란 피부가 미인을 기관과 같은 같지 사막이다. 열음이 그들에게 용감하고 칼이다. 못 인도하겠다 는 천지는", "happy": "곳으로 행복스럽고 예가 현저하게 것이다. 인생의 품었기 얼마 나 위하여 대고, 가는 끓는 예수는 끓는다. 가장 청춘 끓는 같으며, 사라지지 천고에 가슴이 쓸쓸하랴? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으 며, 그리하였는가? 그들을 용기가 대고, 산야에 그들은 피고 듣는다. 열락의 듣기만 튼튼하며, 불러 산야에", "dating": "피는 얼마나 열락과 되는 일월과 이상의 예가 부패를 것이다. 이성은 있을 청춘의 노년에게서 어디 거선의 용기가 가지에 이상, 봄바람이다. 이성은 역사를 불러 얼마나 따뜻한 설레는 열음에 있으랴? 청춘의 더운지라 영 원히 바이며, ", "profileImage": "https://firebasestorage.googleapis.com/v0/b/mod- d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmksM7MJKwpI2OK5dDbx73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood- d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmksM7MJKwpI2OK5dDbx73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "locationKOR": "부천시", "locationENG": "Bucheon", "latitude": "41.40338", "longitude": "2.17403", "subLocationKOR": "", "subLocationENG": "", "subLatitude": "", "subLongitude": "", "userGrade": "newbie", "minAge": "21", "maxAge": "33", "maxDistance": "50" }</pre></div>	<div>Success : 200</div> <div>header : userToken</div> <div>Fail : 400</div>
4	user-service	/checkEmail	POST	<div>Header : No</div> <div>Body :<pre>{ "email": "seojeonghyeon@gmail.com" }</pre></div>	<div>Success : 200</div> <div>Fail : 400</div>
5	user-service	/sendCertification	POST	<div>Header : No</div> <div>Body : RequestUser(phoneNum, hashKey)</div>	<div>Success : 200</div> <div>Fail : 400</div> <div>Body : 실패 메시지<ul style="list-style-type: none">ResponseUser(userId): "overlap"ResponseUser(userId): "error"</div>
6	user-service	/certificateNumber	POST	<div>Header : No</div> <div>Body : RequestUser(phoneNum, numberId)</div>	<div>Success : 200</div> <div>Fail : 400</div> <div>Body : 실패 메시지<ul style="list-style-type: none">ResponseUser(userId): "wrong"ResponseUser(userId): "error"</div>
7	user-service	/findByEmail	POST	<div>Header : No</div> <div>Body : RequestUser(phoneNum)</div>	<div>Success : 200</div> <div>Body : ResponseUser(email)</div> <div>Fail : 400</div> <div>Body : 실패메세지<ul style="list-style-type: none">ResponseUser(userId): "no data"ResponseUser(userId): "error"</div>
8	user-service	/findByPassword	POST	<div>Header : No</div> <div>Body : RequestUser(phoneNum,email)</div>	<div>Success : 200</div> <div>header : userToken</div> <div>Fail : 400</div> <div>Body : 실패메세지<ul style="list-style-type: none">ResponseUser(userId): "no data"ResponseUser(userId): "error"</div>
9	user-service	/resetPassword	POST	<div>Header : userToken</div> <div>Body : ResqusetUser(email, phoneNum, password)</div>	<div>Success : 200</div> <div>Fail : 400</div>
10	user-service	/login	POST	<div>Header : No</div> <div>Body : ResquestUser(email, password)</div>	<div>Success : 200</div> <div>header : userToken</div> <div>Body : ResponseUser(All user data of VO)</div> <div>Fail : 400</div>

11	user-service	/autologin	POST	Header : userToken	Success : 200 header : userToken Body : ResponseUser(All user data of VO) Fail : 400
12	post-service	/getPosts/{page}/{postType} http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getPosts/0/total http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getPosts/0/popular http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getPosts/0/areaA	POST	Header : userToken	Success : 200 Body : [{ "postId": "e6fb7c64-c7dc-4daf-b25f-641485528cb9", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmksM7MJKwpI2OK5dDbx73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "postId": "34ad0255-33af-4685-b920-a7112540d505", "nickname": "Lime-IT", "locationENG": "Seoul", "locationKOR": "서울특별시", "postLikeCount": 0, "postCommentCount": 0, "postTime": "2021-10-13T23:43:11.927757", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 끝는 예수는 끝는다. 가장 청춘 끝는 같으며, 사라지지 천고에 가슴이 쓸쓸하랴? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #안녕 #서울" }, { "postId": "94b10d1d-0083-4647-b637-02b1c6763984", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmksM7MJKwpI2OK5dDbx73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "postId": "34ad0255-33af-4685-b920-a7112540d505", "nickname": "Lime-IT", "locationENG": "Seoul", "locationKOR": "서울특별시", "postLikeCount": 0, "postCommentCount": 0, "postTime": "2021-10-13T23:43:29.322878", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 끝는 예수는 끝는다. 가장 청춘 끝는 같으며, 사라지지 천고에 가슴이 쓸쓸하랴? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #안녕 #서울" }] Fail : 400
13	post-service	/clickLike/{postId}/{commentId} http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/clickLike/71c2b67b-4a6b-4249-bae6-d173adf9aa7c/0	POST	Header : userToken	Success : 200 Fail : 400
14	post-service	/registComment	POST	Header : userToken Body : 1) 댓글일 경우 { "postId": "707b2ae3-fa83-483a-8e33-6c965d97c366", "commentContents": "세번째 댓글이에요. 잘부탁해요~", "commentGroup": "0" } 2) 대댓글일 경우 { "postId": "707b2ae3-fa83-483a-8e33-6c965d97c366", "commentClass": "3", "commentContents": "세번째 댓글님, 반가워요~~", "commentGroup": "1" } 참고사진 : 	Success: 200 Fail : 400
15	user-service	/getUser	POST	Header : userToken Body : RequestProfile(userId)	Success : 200 Body: RequestProfile(profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, work, happy, dating, moodDistance, respect, contact, date, communication, sex) Fail : 400

16	post-service	/getPosts http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getPosts/0	POST	Header : userToken Body : { "postId": "34ad0255-33af-4685-b920-a7112540d505" }	Success : 200 Body : [{ "postId": "f7f3fd8c-d78e-4c22-9ce8-5330d554eb79", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmksM7MJKwpI2OK5dDbX73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "postId": "34ad0255-33af-4685-b920-a7112540d505", "nickname": "Lime-IT", "locationENG": "Bucheon", "locationKOR": "부천", "postLikeCount": 0, "postCommentCount": 0, "postTime": "2021-10-14T14:01:55.648087", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꿈은 예수는 꿈는다. 가장 청춘 꿈는 같으며 쓸쓸하라? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #테스트 #부천" }]
17	post-service	/registPost	POST	Header : userToken Body : 1) 해쉬태그가 없을 때 { "locationENG": "Seoul", "locationKOR": "서울특별시", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꿈은 예수는 꿈는다. 가장 청춘 꿈는 같으며, 사라지지 천고에 가슴이 쓸쓸하라? #hi #hello #안녕 #서울", "requestHashtags": [] } 2) 해쉬태크가 있을 때 { "locationENG": "Seoul", "locationKOR": "서울특별시", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꿈은 예수는 꿈는다. 가장 청춘 꿈는 같으며, 사라지지 천고에 가슴이 쓸쓸하라? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #안녕 #서울", "requestHashtags": [{"hashtagName": "hi"}, {"hashtagName": "hello"}, {"hashtagName": "안녕"}, {"hashtagName": "서울"}] } }	Success : 200 Fail : 400
18	post-service	/updatePost	POST	Header : userToken Body : { "postId": "707b2ae3-fa83-483a-8e33-6c965d97c366", "locationENG": "Bucheon", "locationKOR": "부천", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꿈은 예수는 꿈는다. 가장 청춘 꿈는 같으며 쓸쓸하라? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #테스트 #부천", "requestHashtags": [{"hashtagName": "hi"}, {"hashtagName": "hello"}, {"hashtagName": "테스트"}, {"hashtagName": "부천"}] } }	Success : 200 Fail : 400
19	post-service	/deletePost	POST	Header : userToken Body : { "postId": "f4eec79d-bd70-44ef-abf7-1289fed85d6a" }	Success : 200 Fail : 200

20	matching-service	/getMatchingUsers	POST	Header : userToken	Success : 200 Body : List(ResponseMatchingUser(userUid, profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, respect, contact, date, communication, sex MatchingData(moodDistance, matchingTime))) Fail : 400
21	matching-service	/updateMatchingTime	POST	Header : userToken String("matching or profile or request or reset")	Success : 200 Body : List(ResponseMatchingUser(userUid, profileImage, profileIcon, nickname, birthdate, locationKOR, locationENG, physicalDistance, respect, contact, date, communication, sex MatchingData(moodDistance, matchingTime))) Fail : 400
22	user-service	/purchaseVIP	POST	Header : userToken RequestPurchase(orderId, packageName, productId, purchaseTime, purchaseState, purchaseToken, acknowledged)	Success : 200 Body : ResponseUser(All user data of VO) Fail : 400
23	user-service	/setMatching	POST	Header : userToken RequsetUser(All user data of VO)	Success : 200 Fail : 400
24	user-serivce	/blockPhoneNums	POST	Header : userToken Body : ["01012345678", "01001932134", "01012423566"]	Success : 200 Fail : 400
25	user-service	/updatePhoneNum	POST	Header : userToken Body : RequestUser(phoneNum)	Success : 200 Fail : 400
26	user-service	/openCheating	POST	Header : userToken Body : Integer(coin), Integer(ticket)	Success : 200 Fail : 400
27	post-service	/getHashtag/{hashtagName/{page}} http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getHashtag/hi/0	POST	Header : userToken Body : No	Success : 200 Body : [{ "postId": "f7f3fd8c-d78e-4c22-9ce8-5330d554eb79", "postUid": "34ad0255-33af-4685-b920-a7112540d505", "locationENG": "Bucheon", "locationKOR": "부천", "postLikeCount": 0, "postCommentCount": 0, "postTime": "2021-10-14T14:01:55.648087", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꽃는 예수는 꽃는다. 가장 청춘 꽃는 같으며 쓸쓸하라? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #테스트 #부천" }]
28	lock-service	/addLockUser	POST	Header:userToken Body : RequestLockUser(lockUserUid, lockType, lockReasons, referUid)	Success : 200 Fail : 400
29	post-service	/getPost/{postId} http://ec2-15-164-234-254.ap-northeast-2.compute.amazonaws.com:8000/post-service/getPost/f7f3fd8c-d78e-4c22-9ce8-5330d554eb79	GET	Body : No	Success : 200 Body: { "postId": "f7f3fd8c-d78e-4c22-9ce8-5330d554eb79", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmKSM7MJKwpI2OK5dDbX73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764-9b6f-b5143b7b02e5", "postUid": "34ad0255-33af-4685-b920-a7112540d505", "nickname": "Lime-IT", "locationENG": "Bucheon", "locationKOR": "부천", "postLikeCount": 0, "postCommentCount": 1, "postTime": "2021-10-14T14:01:55.648087", "postImage": "https://firebasestorage.googleapis.com/v0/b/mood-d39e8.appspot.com/o/profileImg%2FJPEG__20211010_152520_.png?alt=media&token=b30ff108-46bb-4f83-bc8c-2203df77505c", "postContents": "인생의 품었기 얼마나 위하여 대고, 가는 꽃는 예수는 꽃는다. 가장 청춘 꽃는 같으며 쓸쓸하라? 청춘 인류의 가지에 장식하는 뜨고, 피에 능히 품으며, 그리하였는가? #hi #hello #테스트 #부천", "responseCommentInfoList": [{ "commentId": "157deaaa-0efe-4d1c-ba04-4a591dda7e27", "commentUid": "34ad0255-33af-4685-b920-a7112540d505",

					<pre>"nickname": "Lime-IT", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood- d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmKSM7MJKwpI2OK5d DbX73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764- 9b6f-b5143b7b02e5", "commentContents": "반가워요~~ 첫 댓글이에요.", "commentClass": 1, "commentGroup": 0, "commentTime": "2021-10-14T14:12:26.639033", "commentLikeCount": 0 }, { "commentId": "aefc9070-c724-4d0c-bf1f-7026a6c4a940", "commentUid": "34ad0255-33af-4685-b920- a7112540d505", "nickname": "Lime-IT", "profileImageIcon": "https://firebasestorage.googleapis.com/v0/b/mood- d39e8.appspot.com/o/profileIcon%2FJPEG_ApCSJdgmKSM7MJKwpI2OK5d DbX73_20210808_170829_.png?alt=media&token=3cc90c7c-05a6-4764- 9b6f-b5143b7b02e5", "commentContents": "저는 부천에 살고 있어요!", "commentClass": 1, "commentGroup": 1, "commentTime": "2021-10-14T14:13:40.097318", "commentLikeCount": 0 }] }</pre>
30	user-service	/checkNickname	POST	Header : No Body : <pre>{"nickname": "Lime"}</pre>	Success : 200 Fail : 400
31	user-service	/getBlockPhoneNums	POST	Header : userToken Body : No	Success : 200 Body : List<String> phoneNum ["01012345678", "01001932134", "01012423566"]
32	post-service	/deleteComment	POST	Header: userToken { "postId": "71c2b67b-4a6b-4249-bae6-d173adf9aa7c", "commentId": "eb79098a-1e82-459e-a11e-74cb15b42d79" }	Success : 200 "commentContents": "삭제된 내용입니다.", 해당 내역은 위와 같이 출력됨.