

FTER API		HOST: http://52.78.166.21:3000 [서울서버]					
	part구분 [디자인 : design , 개발 : develop , 경영마케팅 : bm]		카테고리 구별 [고민있어요 : 1 , 궁금해요 : 2 , 일상이야기 : 3 , 함께해요 : 4]				
설명	메소드	URL	request	response	완성도	비고	수정
로그인 API							
로그인	POST	/login	body: { string id }	첫 로그인 or 프로필 없을 때 : { result: {}, message: 'new' } 이미 등록된 사용자일 때 : { result: { string nickname , string part } , message: 'old' }	100%		1차 : id 암호화 2차 : res에서 result 배열로 넘겨주던거 배열 뺀고 객체로만 넘겨주도록 바꿈 3차 : new일때도 배열알고 객체로 넘겨줌
닉네임 중복 체크	POST	/login/nickcheck	body: { string nickname }	사용할 수 있는 닉네임일 때 : { message: 'true' } 이미 등록된 사용자일 때 : { message: 'false' }	100%		
프로필 작성	POST	/login/profile	body: { string id, string nickname, string part, string statemessage, file image }	재대로 등록됐으면 { message: 'ok' }	100%		1차 : 이미지 1장에서 2장 받아오는걸로 req부분에 image가 2장임 2차 : 이미지 다시 1장으로
타일라인 API							
타일라인 최신순(파트)	GET	/main/partlatest/user_nick/part	params: { string user_nick, string part }	{ result : { { postinfo: { string part, integer level, string nickname, uri profile, string statemessage, integer id, string title, string contents, integer likecount, integer commentcount, string written_time, integer likecheck, integer markcheck, image: { uri image0, uri image1, uri image2, uri image3, uri image4 } } , commentinfo: [{ string user_nick, string content, file image, integer level, string written_time, string statemessage } , { string user_nick, string content, file image, integer level, string written_time, string statemessage }] } } , , message: 'ok' }	100%		1차 : 이미지 10장받아오기에서 다시 5장 받아오기 2차 : commentinfo 넘길 때 댓글을 단 user의 level과 상태메세지 정보 같이넘겨주기
타일라인 인기순(파트)	GET	/main/partpopular/user_nick/part	params: { string user_nick, string part }	{ result : [{ postinfo: { string part, integer level, string nickname, uri profile, string statemessage, integer id, string title, string contents, integer likecount, integer commentcount, string written_time, integer likecheck, integer markcheck, image: { uri image0, uri image1, uri image2, uri image3, uri image4 } } , commentinfo: [{ string user_nick, string content, file image, integer level, string written_time, string statemessage } , { string user_nick, string content, file image, integer level, string written_time, string statemessage }] } } , , message: 'ok' }	100%		1차 : 이미지 10장받아오기에서 다시 5장 받아오기 2차 : commentinfo 넘길 때 댓글을 단 user의 level과 상태메세지 정보 같이넘겨주기
타일라인 최신순(카테고리)	GET	/main/categorylatest/user_nick/category	params: { string user_nick, integer category }	{ result : [{ postinfo: { string part, integer level, string nickname, uri profile, string statemessage, integer id, string title, string contents, integer likecount, integer commentcount, string written_time, integer likecheck, integer markcheck, image: { uri image0, uri image1, uri image2, uri image3, uri image4 } } , commentinfo: [{ string user_nick, string content, file image, integer level, string written_time, string statemessage } , { string user_nick, string content, file image, integer level, string written_time, string statemessage }] } } , , message: 'ok' }	100%		1차 : 이미지 10장받아오기에서 다시 5장 받아오기 2차 : commentinfo 넘길 때 댓글을 단 user의 level과 상태메세지 정보 같이넘겨주기
타일라인 인기순(카테고리)	GET	/main/categorypopular/user_nick/category	params: { string user_nick, integer category }	{ result : [{ postinfo: { string part, integer level, string nickname, uri profile, string statemessage, integer id, string title, string contents, integer likecount, integer commentcount, string written_time, integer likecheck, integer markcheck, image: { uri image0, uri image1, uri image2, uri image3, uri image4 } } , commentinfo: [{ string user_nick, string content, file image, integer level, string written_time, string statemessage } , { string user_nick, string content, file image, integer level, string written_time, string statemessage }] } } , , message: 'ok' }	100%		1차 : 이미지 10장받아오기에서 다시 5장 받아오기 2차 : commentinfo 넘길 때 댓글을 단 user의 level과 상태메세지 정보 같이넘겨주기
좋아요	GET	main/postlike/user_nick/post_id	params: { string user_nick, integer post_id }	좋아요일 때 : { result : [{ integer likecount }], message : 'like' } 좋아요 취소일 때 : { result : [{ integer likecount }], message : 'unlike' }	100%		
북마크	GET	/main/bookmark/user_nick/post_id	params: { string user_nick, integer post_id }	북마크 할 때: { message : 'mark' } 북마크 취소할 때: { message : 'unmark' }	100%		
검색	POST	/main/find	body: { string search_content, string part }	{ result: [{ string user_nick, string title, integer id } ,], message: 'ok' }			

게시글 API							
게시글 작성 화면	GET	/post/write/user_nick	params: { string user_nick }	{ result : [{ uri profile , integer level, string nickname, string part }], message : 'ok' }	100%		
게시글 작성	POST	/post/add	body: { integer category, string part, string title, string contents, file image, string user_nick }	{ message : 'ok' }	100%	req의 image는 최대 5장까지 array로 넣을 수 있음	
게시글 자세히 보기	POST	/post/read	body: { integer post_id, string user_nick }	{ result: { postinfo: { string nickname , integer level , string userpart , uri profile, string statemessage, string title, string contents, string written_time, string postpart, integer category, integer likecount, integer commentcount, integer likecheck, integer markcheck, image: [uri image0, uri image1, uri image2, uri image3, uri image4] }, commentinfo: [{ string user_nick, string content, uri image, string written_time, integer level, string part, string statemessage } ,] } , message : 'success' }	100%		1차 : res에서 postinfo와 commentinfo를 가져올 때 level과 statemessage 정보 같이 넘겨주기 2차 : 댓글이 없을 때 null 이 넘어가는거 그냥 비어있도록 수정
게시글 삭제 검사	POST	/post/deletecheck	body: { integer post_id, string user_nick }	거절! { message: 'deny' } 확인! { message: 'ok' }	100%	삭제 라는 버튼이 보이게 할 지 안할 지 유효성 검사	
게시글 삭제	POST	/post/delete	body: { integer post_id, string user_nick }	{ message: 'success' }	100%	삭제 검사를 위해서 거치니 이 부분에서는 따로 안함	
댓글 API							
댓글 전체보기[최신순]	GET	/comment/post_id	params: { integer post_id }	{ result: [{ string user_nick, string content, integer level, string statemessage, uri image, string written_time } ,], message : 'ok' }	100%		1차 : res에서 댓글 정보 asc에서 desc로 수정
댓글 작성	POST	/comment/add	body: { string user_nick, integer post_id, integer useful, string content }	{ message: 'ok' }	100%		
유용한 댓글 보기	GET	/comment/useful/post_id	params: { integer post_id }	{ result: [{ string user_nick, string content, integer level, string statemessage, uri image, string written_time } ,], message: 'ok' }	100%		1차 : res에서 댓글 정보 asc에서 desc로 수정
마이페이지 API							
마이페이지	GET	/mypage/user_nick	params: { string user_nick }	{ result: { integer level, uri profile, string nickname, string statemessage, string part } , message: 'ok' }	100%		
내가 찜한글	GET	/mypage/like/user_nick	params: { string user_nick }	{ result: [{ string title, string written_time, integer id } ,], message: 'ok' }	100%		
내가 쓴글	GET	/mypage/write/user_nick	params: { string user_nick }	{ result: [{ string title, string written_time, integer id } ,], message: 'ok' }	100%		
개인정보 수정	POST	/mypage/edit	body: { string user_nick, string nickname, string part, string statemessage, file image }	{ message: 'update' }	99%	정보를 모두바꾸는게 아니라 하나 또는 두개만 바꿀 때 통신이 어떻게 이루어져야하는지 클라량 연동할 때 확인	1차수정 : 프로필작성과 마찬가지로 이미지 2개받아옴, 첫 file image는 작은거 두번째 file image는 큰거 2차수정 : image는 한장만 받아오기 3차수정 : 닉네임은 안바꿀때 예외처리 추가
닉네임 중복 체크(수정)	POST	/mypage/nickcheck	body: { string oldnick, string newnick }	사용할 수 있는 닉네임 or 기존 닉네임일 때 : { message: 'true' } 이미 등록된 사용자일 때 : { message: 'false' }	100%		

알람 API					100%		
알람창 보기	GET	/alarm/user_nick	params: { string user_nick }	{ integer count, result: { { integer id, integer readinfo, string written_time, integer postid, string title }, }, message: 'ok' }	100%		
해당 알람 클릭	GET	/alarm/readinfo/alarm_id	params: { integer alarm_id }	{ message : 'ok' }	100%		1차수정 : 해당 게시글의 댓글 모두보기로 가던거를 해당 게시글 자체히 보기로 수정