

API DOCS												
path0	path1	path2	path3	method	설명	header	params	body	return	완성	담당자	비 고
api/	categories	-	-	GET	학습 전용 카테고리 목록을 리스트 배열에 담아 응답한다.	-	-	-	{ categoryId : 1, name: '자음', isTestable: 'T', ... }	O	손민혁	
api/	words	-	-	GET	특정 조건을 만족하는 모든 단어를 리스트에 담아 응답한다.	-	categoryId (선택), isRandom (선택), isTestable (선택), num (선택)	-	{ id: '1', name: 'ㄱ', contentUrl: 'www.cloud.com', ... }	O	손민혁	categoryId는 category의 PK입니다. isRandom=true 를 붙여서 요청을 보내면 단어 순서를 랜덤으로 섞어서 응답을 보냅니다. num에 원하는 단어 개수를 입력하면 원하는 개수의 단어를 응답합니다. isTestable=false 를 입력하면 테스트 불가능한 단어들을 요청합니다. 응답 예시) 전체 단어가 필요할 때 -> /api/words 1번 카테고리에 속하는 단어들이 필요할 때 -> /api/words?category=1 1번 카테고리에 속하는 랜덤 순서의 단어들이 필요할 때 -> /api/words?category=1&isRandom=true 랜덤한 12개의 단어가 필요할 때 -> /api/words?isRandom=true&isTestable=false&num=12
api/	favorites	-	-	GET	나의 단어장에 등록된 단어들을 리스트에 담아 응답한다.	Authorization : access token	isRandom (선택)	-	{ id: '1', name: 'ㄱ', contentUrl: 'www.cloud.com', categoryId: '1', ... }	O	손민혁	예시) 단순 조회 시 api/favorites 예시) 서플 기능 추가 시 api/favorites?isRandom=true
api/	favorites/	word/	{wordId}	GET	나의 단어장에 있는 단어인지 확인한다. 성공 시 "success", 실패 시 에러코드 404 와 에러 메시지 를 반환한다.	Authorization : access token	wordId	-	{ message: "success" }	O	손민혁	wordId는 word의 PK입니다. 예시) api/favorites/word/1
api/	favorites	-	-	POST	특정 단어를 나의 단어장에 추가한다. 성공 시 "success", 실패 시 에러코드 404 와 에러 메시지 를 반환한다.	Authorization : access token	-	{ wordId: wordId }	{ message: "success" }	O	손민혁	wordId는 word의 PK입니다. 예시) POST api/favorites
api/	favorites	-	-	DELETE	특정 단어를 나의 단어장에 삭제한다. 성공 시 "success", 실패 시 에러코드 404 와 에러 메시지 를 반환한다.	Authorization : access token	-	{ wordId: wordId }	{ message: "success" }	O	손민혁	wordId는 word의 PK입니다. 예시) DELETE api/favorites
api/	user/	signup/	email/ {email}	GET	일반 회원 회원가입 시 email 비동기로 중복 체크 성공 시 (중복되지 않은 경우) "success" 실패 시 (중복된 경우) 에러코드 409 와 에러메시지 를 반환한다.	-	email	-	{ "message": "success" }	O	황정주	
api/	user/	signup/	nickname/ {nickname}	GET	일반 회원 회원가입 시 nickname 비동기로 중복 체크 성공 시 (중복되지 않은 경우) "success" 실패 시 (중복된 경우) 에러코드 409 와 에러메시지 를 반환한다. nickname : 2자 이상, 10자 이하 문자열	-	nickname	-	{ "message": "success" }	O	황정주	
api/	user/	signup	-	POST	일반 회원 회원가입 페이지에서 가입하기 버튼 눌렀을 때 성공 시 "success"와 200 OK , 실패 시 에러 메시지 를 반환한다. userType : "NORMAL" - 일반 회원, "KAKAO" - 카카오톡 회원 password : 8자 이상 20자 이하 문자열 nickname : 2자 이상, 10자 이하 문자열	-	-	{ userType : userType, email : id, password : password, nickname : nickname, picture : profile picture URL }	{ "message": "success" }	O	황정주	
api/	user/	login	-	POST	일반 회원 로그인 password : 8자 이상, 20자 이하 문자열 성공 시 바디에 회원 정보와 "success"를 , 헤더에 refreshToken을 담은 쿠키를 담는다. 실패 시 에러코드 404 와 에러메시지 를 반환한다.	-	-	{ email : id, password : password }	{ "nickname": "nickname", "picture": "profile picture URL", "level": "level", "experience": "experience", "accessToken": "트렌값", "message": "success" }	O	황정주	
api/	user/	logout	-	GET	일반 회원 카카오톡 회원 로그아웃 성공 시 "success", 실패 시 에러 메시지 를 반환한다.	Authorization : access token	-	-	{ "message": "success" }	O	황정주	

API DOCS												
path0	path1	path2	path3	method	설명	header	params	body	return	완성	담당자	비 고
api/	user/	profile	-	PUT	프로필 수정 성공 시 200코드 와 "success" , 실패 시 에러 메시지 를 반환한다.	Authorization : access token	-	{ nickname : nickname, picture : profile picture URL }	{ "success" }	O	황정주	
api/	user/	profile	-	GET	회원(프로필) 정보 조회 성공 시 200코드 와 email(일반 사용자 id), kakaoid(카카오 사용자 id)를 모두 포함한 사용자 정보 를 리턴한다. 클라이언트쪽에서는 토큰만 갖고 있기 때문에 현재 userType(NORMAL or KAKAO)를 모른다. 따라서 email, kakaoid 모두 담아서 보낸다.	Authorization : access token	-	-	{ "id" : "user 테이블의 PK" "userType" : user Type, "email" : "email", "kakaoid" : "kakaoid", "nickname" : "nickname", "picture" : "profile picture URL", "level" : "level", "experience" : "experience" "accessToken" : "토큰값" }	O	황정주	
api/	user/	game/	{experience}	PUT	게임, 테스트 결과 경험치 반영 성공 시 200코드 와 결과가 반영된 level, experience 를 리턴한다.	Authorization : access token	-	{ experience : 경험치 }	{ "level" : "level" "experience" : "experience" }	O	황정주	GA06 (게임 결과창)이 뜨는 시점에 요청을 한다.
api/	user/	refresh	-	GET	Access Token 재발급 성공 시 200 코드 와 "success" , 실패 시(refreshToken 이 유효하지 않은 경우) 401 코드와 에러메시지 를 리턴한다.	Authorization : access token	-	-	{ "message" : "message", "accessToken" : "토큰값" }	O	황정주	
api/	user/	oauth2/	kakao	GET	카카오 회원 가입 카카오 로그인 인가코드를 파라미터로 받아와서 카카오로부터 user 정보를 받아오고 가공하여 user 정보를 반환한다. 성공 시 "success"와 user 정보 , 실패 시 401 코드와 에러메시지 를 리턴한다.	-	code	-	{ "nickname" : "nickname", "picture" : "profile picture URL", "level" : "level", "experience" : "experience" "accessToken" : "토큰값", "message" : "success" }	O	황정주	
api/	user/	ranking		GET	실시간 랭킹 경험치를 기준으로 상위 10명의 데이터 를 반환한다.				{ "nickname" : "nickname", "picture" : "picture", "level" : "level", "experience" : "experience" }	O	황정주	
api/	game/	session	-	POST	현재 대기방 session 에 유저를 연결하기 위해 connection 을 생성한다. 만약 session 의 정원이 채워지면 게임을 시작하라는 값을 설정한다 (playGame=true). 생성된 connection 의 token 값, 현재 session 에 연결된 유저 리스트, 게임 시작 여부 등을 반환한다.	Authorization : access token	-	-	{ "message" : "success" "sessionId" : sessionId "token" : "토큰값" "playGame" : true/false "playerList" : [] "imageList" : [] }	O	임영목	FE에서 토큰을 받아서 publish하면, 연결되어있는 모든 client들이 "streamCreated" 이벤트가 발생되면서 콜백함수가 호출된다.
api/	game/	session/	user	POST	대기방에서 유저가 입장 시 호출되어, 서버의 자료구조에서 해당 내용을 동기화한다. 만약 현재 대기방에서 마지막 유저가 나갈다면 OpenVidu 정책 상 session 이 자동으로 닫히기 때문에 현재 대기방을 폐기하고 새로운 대기방으로 교체한다.	Authorization : access token	-	{ sessionId : sessionId }	{ "message" : "success" }	O	임영목	
api/	game/	session/	{sessionId}	DELETE	해당 게임 session 을 종료시키고, 서버의 자료구조에서 해당 session 을 제거하고, DB에 게임 로그를 저장한다.	Authorization : access token	-	-	{ "message" : "success" }	O	임영목	
api/	game/	session/	session	PUT	현재 대기방에 있는 모든 유저들을 되돌리고 대기방을 초기화한다.	Authorization : access token	-	-	{ "message" : "success" }	O	임영목	
ws/	socket-server/	{categoryId}	-	GET	특정 카테고리에 대한 손동작 인식 결과를 반환한다.	-	-	-	{ "response": '...' }	O	손민혁	