//////// 모든 에러

[response]

- status : 404

- message : "상황에 맞는 어디서 에러 났는지 메시지 출력"

- data : null

/////// 회원가입, 로그인, 로그아웃

-- 회원가입, 로그인

POST (/member)

[request]

- kakao\_id : 회원식별번호

- nickname : 닉네임

- profile : 프로필 사진 파일

- address : 주소

- latitude : 위도

- longitude : 경도

[response]

- status : 201

- message : "회원가입 성공"

- data : null

-- 로그아웃

POST (/member)

[request]

- kakao\_id : 회원식별번호

[response]

- status : 200

- message : "로그아웃 성공"

- data : null

///////// 프로필

- 프로필 정보 불러오기

GET (/member)

[response]

- status : 200

- message : "프로필 정보 가져오기"

- data : {

nickname : 닉네임

profile : 프로필 사진 url

total\_count : 거래 수

sugar\_score : 당도

time\_good : 시간 어쩌고

nice\_guy : 친절해요

food\_divide : 음식 나눠

fast\_answer : 응답 빨라

reviews : [ -> 최대 3개까지만 보여주고 나머지는 따로 요청

{

review : "정말 좋았어요",

sender : "지누"

}, {

review : "정말 극혐이에요".

sender : "누군가"

}

]

}

- 프로필 창 수정

PUT (/member)

[request]

- nickname : 닉네임

- profile : 프로필 사진 파일

[response]

- status : 200

- message : "프로필 수정 완료"

- data : null

- 활동 목록

GET (/done\_post)

[response]

- status : 200

- message : "활동 목록 불러오기 성공"

- data :

{

posts : [

{

postId : 3,

title : "공차 버블티 같이 드실분",

host : true / false

}

]

}

- 받은 후기

GET (/reviews)

[response]

- status : 200

- message : "받은 후기 불러오기 성공"

- data :

{

reviews : [ -> 전체 리뷰 불러오기

{

review : "정말 좋았어요",

sender : "지누"

}, {

review : "정말 별로에요".

sender : "누군가"

}

]

}

////////// 필터 설정

////

Korean = 0;

Japanese = 1;

SchoolFood = 2;

Dessert = 3;

Chicken = 4;

Pizza = 5;

Western = 6;

Chinese = 7;

NightFood = 8;

FastFood = 9;

////

PUT (/filter)

[request]

- distance : 반경

- category : [0, 5, 7] -> Korean, Pizza, Chinese를 선택했다는 뜻

[response]

- status : 200정

- message : "필터 수 완료"

- data : null

///////// 후기 작성

POST (/review)

[request]

- preference : 최고에요(2), 좋아요(1), 별로에요(0) 클릭한 버튼에 따라 해당 숫자를 넣어주면 됨

- time\_good : true

- nice\_guy : false

- food\_divide : false

- fast\_answer : true

[response]

- status : 201

- message : "후기 작성 완료"

- data : null

게시글 (post/)

////////// 게시글 작성

////

Korean = 0;

Japanese = 1;

SchoolFood = 2;

Dessert = 3;

Chicken = 4;

Pizza = 5;

Western = 6;

Chinese = 7;

NightFood = 8;

FastFood = 9;

////

POST (/post)

[request]

- title : 제목

- description : 내용

- chatLink : 채팅 링크

- foodLink : 음식뭐 그거 링크

- category : 0 -> 카테고리

- memberCountLimit : 인원 제한

- orderTime : 주문 시간

- meetPlace : 만날 장소(0 : 호스트, 1 : 중간지점, 2 : 파티원)

- deliveryFeeByHost : 배달비(0 : 각자 부담, 1 : 호스트 부담)

[response]

- status : 201

- message : "게시글 작성 완료"

- data : null

////////// 게시글 수정

PUT (/post)

[request]

- title : 제목

- description : 내용

- chatLink : 채팅 링크

- foodLink : 음식뭐 그거 링크

- category : 0 -> 카테고리

- memberCountLimit : 인원 제한

- orderTime : 주문 시간

- meetPlace : 만날 장소(0 : 호스트, 1 : 중간지점, 2 : 파티원)

[response]

- status : 200

- message : "게시글 수정 완료"

- data : null

////////// 게시글 삭제

DELETE (/post/{postId})

[response]

- status : 200

- message : "게시글 삭제 완료"

- data : null

////////// 게시글 검색

////

Korean = 0;

Japanese = 1;

SchoolFood = 2;

Dessert = 3;

Chicken = 4;

Pizza = 5;

Western = 6;

Chinese = 7;

NightFood = 8;

FastFood = 9;

////

GET (/post?distance={dist}&category=0-2-4-6&page=0&size=10)

- distance, cateogry 둘 다 있으면 해당 필터 적용된 게시글만 검색.

- distance 없으면 범위 제한 없이 게시글 다 불러옴(거리 까가운 순으로 정렬해서)

- page : size 기준으로 몇 번째 페이지인지(0부터 시작)

- size : 게시글 몇 개를 불러올 것인지

--> category, page, size는 필수

[response]

- status : 200

- message : "게시글 목록 불러오기 완료"

- data : {

posts : [

{

title : 제목,

foodLink : 일단 둬,

deliveryFeeByHost : 누가 부담할지,

meetPlace : 만날 장소(0 : 호스트, 1 : 중간지점, 2 : 파티원)

memberCount : 2,

categoryImage : "https://어쩌고.컴",

memberCountLimit : 4,

orderTime : 배달 시간

distance : 거리,

isFavorite : true/false 찜했는지 여부

}

]

}

///////// 게시글 세부 보기

GET (/post/{postId})

[response]

- status : 201

- message : "게시글 작성 완료"

- data : {

nickname : 닉네임,

profile\_url : 프로필 사진,

sugar\_score : 당도,

foodLink : 음식 주 링크,

chatLink : 오픈채팅 링크,

description : 게시글 상세 글

}

문

////////// 찜하기

GET (/favorite/{postId})

[response]

- status : 201

- message : "게시글 작성 완료"

- data : null