도메인: <https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test>

모든 요청의 Content-Type은 application/json이거나 multipart/form-data입니다.

Type은 명시되지 않으면 String입니다.

1. 버킷리스트 작성:

Path: /bucketlist

Content-Type: Multipart/form-data

Method: POST

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“startDate”: “2022-01-01”

“endDate”: “2022-03-30”,

“title”: \_\_\_\_ ,

“content” : \_\_\_\_\_\_,

“isVisible”: \_\_\_ (integer),

“detailPlans”: “[

{

“orderNumb”: \_\_, 1,2,3,...

“content”: \_\_\_\_\_,

“planDate”: \_\_\_\_

}

]”,

“picuture”: \_\_\_\_(binary data),

“tags”: “["운동", "맛집"]”,

}

* 사진은 picture keyword로 안보내주셔도 됩니다.

Response Body:

{

“status”: \_\_\_\_,

“msg”: \_\_\_\_\_ ,

“bucketList”: {

“bucketId”:\_\_\_\_, (나의 상세 정보 페이지 에서 내것 표시)

“title”: \_\_\_\_,

“content”: \_\_\_\_\_\_,

“isVisible”: \_\_\_\_(integer),

“startDate”: \_\_\_\_\_,

“endDate”: \_\_\_\_\_,

“tags”: [ “\_\_\_”, “\_\_\_\_\_”],

“detailPlans”: [

{

“orderNumb”: \_\_(integer),

“content”: \_\_\_\_\_,

“planDate”: \_\_\_\_

}

],

“profilePict”: \_\_\_\_\_\_\_

}

}

* isVisible 은 0 아니면 1 입니다. 0이면 private 1이면 public
* bucketId는 숫자입니다.

status -> 0일때 버킷리스트 작성 완료, 1일때 작성 실패

bucketList -> status가 0일때 리턴

1. 버킷리스트 수정

Path: /bucketlist

Content-Type: Multipart/form-data

Method: PUT

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“startDate”: “2022-01-01”

“endDate”: “2022-03-30”,

“title”: \_\_\_\_ ,

“content” : \_\_\_\_\_\_,

“isVisible”: <0 or 1>,

“detailPlans”: “[

{

“orderNumb”: \_\_(integer),

“content”: \_\_\_\_\_,

“planDate”: \_\_\_\_

}

]”,

“picuture”: \_\_\_\_\_ (binary data),

“bucketId”: \_\_\_\_\_(integer),

“tags”: “["운동", "맛집"]”,

}

* 사진은 picture keyword로 안보내주셔도 됩니다.

Response Body:

{

“status”: \_\_\_\_,

“msg”: \_\_\_\_\_ ,

“bucketList”: {

“bucketId”:\_\_\_\_,

“title”: \_\_\_\_,

“content”: \_\_\_\_\_\_,

“isVisible”: \_\_\_\_,

“startDate”: \_\_\_\_\_,

“endDate”: \_\_\_\_\_,

“tags”: [ “\_\_\_”, “\_\_\_\_\_”],

“detailPlans”: [

{

“orderNumb”: \_\_(integer),

“content”: \_\_\_\_\_,

“planDate”: \_\_\_\_

}

],

“profilePict”: \_\_\_\_\_\_\_

}

}

* bucketId는 숫자입니다.

status -> 0일때 버킷리스트 수정 완료, 1일때 수정 실패

bucketList -> status가 0일때 리턴

1. 버킷리스트 삭제 가능하게

Path: /bucketlist

Content-Type: Application/json

Method: DELETE

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“bucketId”:\_\_\_(Integer)

}

Response Body:

{

“status”: \_\_\_\_,

“msg”: \_\_\_\_\_ ,

}

* bucketId는 숫자입니다.

status -> 0일때 버킷리스트 삭제 완료, 1일때 삭제 실패

bucketList -> status가 0일때 리턴

1. 버킷리스트 인증가능하게 -> (사진, 인증날짜, 인증여부) 사진으로 인증, 인증이 안될시 알람(사진저장)!

path: [bucketlist/authenticate](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/authenticate)

Method: POST

Content-type: Multipart/form-data

Request Header: {  
 “Authorization”: \_\_\_\_\_(String)

}

Request Body:{

“pict”: \_\_\_\_(binary data),

“bucketId”: \_\_\_(Integer)

}

Response Body:{

“msg”:

“status”:

}

1. 버킷리스트 추천(관심 카테고리에 등록된것, 순서는 생성된 날짜순으로)

Path: [bucketlist/recommendation](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/recommendation)

Content-Type: application/json

Method: GET

Request Header:

{

“Authorization”: <token>

}

* Request Body 필요 없습니다.

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_\_\_,

"bucketLists": [

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_ ,

"heart": \_\_\_\_ (integer),

“achieved”: 0 or 1

},

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_ ,

"heart": \_\_\_\_ (integer),

“achieved”: 0 or 1

}

]

}

* bucketId는 숫자입니다.
* heart -> 0이면 유저가 하트를 누르지 않았다는 뜻이고, 1이면 유저가 하트를 눌렀다.

status -> 0일때 버킷리스트 추천 완료, 1일때 추천 실패

bucketLists -> status가 0일때 리턴

1. 버킷리스트 검색(키워드가 태그와 제목에 매칭이 되어야함)

Path: [bucketlist/search](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/search)

Content-Type: application/json

Method: GET

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“keyword”: \_\_\_\_\_

}

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_\_\_,

"bucketLists": [

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_ ,

"heart": \_\_\_\_ (integer)

},

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_ ,

"heart": \_\_\_\_ (integer)

}

]

}

* bucketId는 숫자입니다.
* heart -> 0이면 유저가 하트를 누르지 않았다는 뜻이고, 1이면 유저가 하트를 눌렀다.

status -> 0일때 버킷리스트 search 완료, 1일때 search 실패

bucketLists -> status가 0일때 리턴

1. 인증이 되어야 하는데 인증이 안되는것 같으면 -> (사진,내용, 사진찍은 날짜, 버킷리스트) 운영자에게 알릴수있는 기능!

Path: [bucketlist/verifyrequest](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)

Content-Type: multipart/form-data

Method: POST

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“bucketId”: \_\_\_\_\_ (integer),

“content”: \_\_\_\_\_,

“picture”: \_\_\_\_\_\_ (binary Data)

}

* 사진은 picture keyword로 안보내주셔도 됩니다.
* 사진을 보내지 않으면 -> fail

Response Body:

{

"status": \_\_\_\_\_\_,

"msg": \_\_\_\_\_\_\_\_\_\_\_,

"verification\_request": {

"bucketId":\_\_\_\_\_(integer) ,

"content": \_\_\_\_,

"pictPath": \_\_\_\_\_ }

}

* bucketId는 숫자입니다.

status -> 0일때 문의 완료, 1일때 문의 실패

1. 버킷리스트의 후기쓰기 -> 사진은 최대 10장

Path: [bucketlist/](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)review

Content-Type: multipart/form-data

Method: POST

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“bucketId”: \_\_\_\_\_ (integer),

“content”: \_\_\_\_\_,

“picture”: \_\_\_\_\_\_ (binary Data)

}

* 사진은 picture keyword로 안보내주셔도 됩니다.
* 사진은 최대 10장까지 가능합니다.

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_ ,

"review": {

"picts": [ \_\_\_\_, \_\_\_\_\_\_ ],

"content": "정말 재밌었습니다.",

"bucketId": "27"

}

}

* bucketId는 숫자입니다.

status -> 0일때 리뷰작성 완료, 1일때 리뷰작성 실패

1. 버킷리스트 상세 조회 -> 제목, 내용, 세부계획, 사진(버킷리스트 프로필 사진), 태그들, 시작날짜, 완료 날짜

Path: [bucketlist](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)

Content-Type: application/json

Method: GET

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

"bucketId": \_\_\_ (integer)

}

Response Body:

{

"status": \_\_\_\_,

"msg": \_\_\_\_\_\_\_\_,

"bucketList": {

"bucketId": \_\_\_\_(integer),

"isVisible": \_\_\_(integer),

"title": \_\_\_\_\_\_\_,

"nickname": \_\_\_\_\_\_\_ ,

"startDate": \_\_\_\_\_\_\_\_,

"endDate": \_\_\_\_\_\_\_\_\_,

"heart": \_(integer),

"tags": [

\_\_\_\_\_\_,

\_\_\_\_\_

],

"profilePict": **\_\_\_\_\_\_\_** ,

"achievedPict": \_\_\_\_\_\_,

"reviewPicts": [

\_\_\_\_\_\_\_\_\_\_(String) ->

],

"detailPlans": [

{

"orderNumb": \_\_\_\_\_(integer),

"content": \_\_\_\_\_\_\_\_,

"planDate": \_\_\_\_\_\_\_\_\_,

"achieved": \_\_\_\_\_\_\_\_\_

},

{

"orderNumb": \_\_\_\_\_(integer),

"content": \_\_\_\_\_\_\_\_,

"planDate": \_\_\_\_\_\_\_\_\_,

"achieved": \_\_\_\_\_\_\_\_\_

}

],

"reviewContent": \_\_\_\_\_\_\_\_\_,

“bucketContent”:\_\_\_\_\_\_\_ ,

“userProfilePict” : \_\_\_\_\_\_\_\_\_

}

}

* bucketId는 숫자입니다.
* achieved -> “0” 이면 완료되지 않은 버킷리스트, “1” 이면 완료된 버킷리스트
* heart -> 0이면 유저가 하트를 누르지 않았다는 뜻이고, 1이면 유저가 하트를 눌렀다.

status -> 0일때 상세보기 성공, 1이면 상세보기 실패

1. 하트 누른 버킷리스트만 따로 조회할수 있게

Path: [bucketlist/](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)search/heart

Content-Type: application/json

Method: GET

Request Header:

{

“Authorization”: <token>

}

Response Body:

{

"status": \_\_\_\_\_,

"msg": \_\_\_\_\_\_,

"bucketLists": [

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": **\_\_\_\_\_\_**,

"heart": \_\_\_\_(integer)

},

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": **\_\_\_\_\_\_**,

"heart": \_\_\_\_(integer)

},

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": **\_\_\_\_\_\_**,

"heart": \_\_\_\_(integer)

}

]

}

* bucketId는 숫자입니다.
* heart -> 0이면 유저가 하트를 누르지 않았다는 뜻이고, 1이면 유저가 하트를 눌렀다.

status -> 0일때 request 성공, request 실패

1. 알람 조회 -> 봤던 알람과 보지 못한 알람을 조회할수 있게!
2. 버킷리스트 후기 요청 알람: (버킷리스트, d-day, 메세지) -> 후기를 써야하는 마감날에 알람! (0)
3. 포인트 적립알람: (버킷리스트, 사진, 포인트, 적립날짜) (1)
4. 버킷리스트 거부알람: (버킷리스트, 사진, 거부알림 날짜) (2)
5. 버킷리스트 미완료시 알람:( 버킷리스트, 몇일남았는지) -> 2일전, 하루전, 당일 (3)
6. 세부계획 미완료시 알람: (버킷리스트, 세부계획) 당일 (4)

12) 유저정보 수정: 프로필사진, 추천 카테고리 설정, 닉네임

:

(토큰, 프로필사진, 태그, 닉네임) ->

성공 or 실패 메시지

Path: user

Content-Type: multipart/form-data

Method: PUT

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“tags”: “[ \_\_\_\_ , \_\_\_\_\_ ]”

"nickname” : \_\_\_\_,

“picture” : \_\_\_\_ (binary data)

}

* 사진은 picture keyword로 안보내주셔도 됩니다.
* 사진을 보내지 않으면 -> fail

Response Body:

{

"status": \_\_\_\_\_,

"msg": \_\_\_\_\_\_,

}

status -> 0일때 request 성공, request 실패

13) 버킷리스트 하트누르기

Path: bucketlist/heart

Content-Type: application/json

Method: POST

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“bucketId”: \_\_\_ (integer)

}

Response Body:

{

"status": \_\_\_\_\_,

"msg": \_\_\_\_\_\_,

}

status -> 0일때 request 성공, request 실패

14) 버킷리스트 하트누르기 취소

Path: bucketlist/heart

Content-Type: application/json

Method: DELETE

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

“bucketId”: \_\_\_ (integer)

}

Response Body:

{

"status": \_\_\_\_\_,

"msg": \_\_\_\_\_\_,

}

status -> 0일때 request 성공, request 실패

15) 유저 포인트 조회

:

Path: user/point

Content-Type: application/json

Method: GET

Request Header:

{

“Authorization”: <token>

}

Response Body:

{

"status": \_\_\_\_\_,

"msg": \_\_\_\_\_\_\_\_\_,

"point": \_\_(Integer)

}

status -> 0일때 request 성공, request 실패

16) 마이 버킷리스트 조회

:

Path: bucketlist/my

Content-type: application/json

Method: Get

Request Header:

{

“Authorization”: <token>

}

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_\_\_,

"bucketLists": [

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_

},

{

"bucketId": \_\_\_\_(integer),

"title": \_\_\_\_\_\_,

"profilePict": \_\_\_\_\_

}

]

}

17) 유저 정보 설정 페이지 불러오기 ( 프로필 사진, 닉네임, 이메일, 관심 카테고리)

Path: /user

Content-type: application/json

Method: Get

Request Header:

Request Header:

{

“Authorization”: <token>

}

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_\_\_,

"user": {  
 “email”: \_\_\_\_\_,

“nickname”: \_\_\_\_\_\_,

“profilePicturePath”: \_\_\_\_\_\_\_\_,

“tags”: [“\_\_\_”, “\_\_\_\_\_”]

}

}

18) 문의하고나서 문의내역 보여주기 (문의한 날짜, 문의한 버킷리스트 제목, 문의할 달성사진, 문의 내용)

Path: [/bucketlist/verifyrequest](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)

Content-type: application/json

Method: Get

Request Header:

Request Header:

{

“Authorization”: <token>

}

Response Body:

{

"status": \_\_\_,

"msg": \_\_\_\_\_\_,

"verifyRequests": [

{

“requestDate”: \_\_\_\_\_,

“title”: \_\_\_\_\_\_\_,

“verifyRequestPicts”:[“\_\_\_\_\_”, “ \_\_\_\_” , “\_\_\_\_\_”]

“content”: \_\_\_\_\_\_\_,

“state”: \_\_(Integer),

“bucketId”: \_\_\_\_(Integer)

},

{

“requestDate”: \_\_\_\_\_,

“title”: \_\_\_\_\_\_\_,

“verifyRequestPicts”:[“\_\_\_\_\_”, “ \_\_\_\_” , “\_\_\_\_\_”]

“content”: \_\_\_\_\_\_\_,

“state”: \_\_\_\_\_(Integer),

“bucketId”: \_\_\_\_(Integer)

}

]

}

state -> 0이면 아직 검토 x, 1이면 reject, 2면 accept

19) 유저 포인트 지급받은 내역 (100 지급시 +100, 100 사용시 -100)

Path: [/user/point/details](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/user/point/details)

Content-type: application/json

Method: Get

Request Header:

{

“Authorization”: <token>

}

Response Body:{

“msg”: \_\_\_\_\_,

“status”: \_\_\_\_\_,

"details": [

{

"source": "review" or “purchase”,

"formatTime": "5/18/2022, 4:13:50 PM"(String),

"point": 2000 (Integer),

“bucketId”: \_\_\_\_\_(Integer),

“title”: \_\_\_\_\_\_(String)

}

]

}

Soure: review

20) 유저 랭킹 가져오기

Content-type: application/json

Path: [/user/rankings](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/user/rankings)

Method: Get

Request Header:

{

“Authorization”: <token>

}

Response Body:{

"status": "0",

"msg": \_\_(String),

"ranking": [

{

"email": "jjieun412@gmail.com",

"point": 1000,

"nickname": "banilaCoff",

"profilePicturePath": "https://buckybucket2022.s3.ap-northeast-2.amazonaws.com/profilepict/jjieun412@gmail.com.jpg"

},

{

"email": "harryjung0330@gmail.com",

"point": 0,

"nickname": "bidd",

"profilePicturePath": **null**

},

{

"email": "jieun@gmail.com",

"point": 0,

"nickname": "micnkyk",

"profilePicturePath": "https://buckybucket2022.s3.ap-northeast-2.amazonaws.com/profilepict/jieun@gmail.com.jpg"

}

],

"myinfo": {

"nickname": "bidd",

"profilePicturePath": **null**

}

}

21) 세부계획 완료/완료 취소:

Content-type: application/json

Method: PUT

Path: [/bucketlist/detailplan/achieved](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/detailplan/achieved)

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

bucketId: (Integer)

orderNumb:(Integer)

achieved: 0 or 1

}

0이면 미성취

1이면 성취

Response Body:

{

“msg”: \_\_\_,

“status”: \_\_\_\_\_

}

22) 문의 내역 삭제:

Content-type: application/json

Method: DELETE

Path: [/bucketlist/verifyrequest](https://5ns0yfs844.execute-api.ap-northeast-2.amazonaws.com/test/bucketlist/verifyrequest)

Request Header:

{

“Authorization”: <token>

}

Request Body:

{

bucketId: (Integer)

requestDate:(String)

}

Response Body:

{

“msg”: \_\_\_,

“status”: \_\_\_\_\_

}