

Daily Q API

v0.1.1

2021. 2. 28

장성환

개정 이력

[illegible]

Version 1

Version 1 은 개발 초기 단계에서 인증 없이 테스트를 할 수 있도록 준비한 API 입니다. 인증 없이 모든 요청에 anonymous 계정이 사용됩니다. Version 1 의 모든 API 는 /v1 으로 시작합니다.

Hello, world!

서버로부터 Hello, world! 문자열을 가져옵니다. 요청의 Accept 헤더에 따라 단순 문자열 혹은 JSON 데이터를 받습니다.

요청

GET /v1/hello-world

이름	위치	타입	필수	설명
Accept	header	string		- text/plain (기본) - application/json

응답

200: 성공

Accept: text/plain

```
Hello, world!
```

Accept: application/json

```
{
  "message": "Hello, world!",
  "date": "2021-05-01T19:36:27Z"
}
```

이름	타입	설명
message	string	Hello, world! 문자열
date	string	요청 날짜와 시간

질문 가져오기

경로로 전달된 질문 아이디(특정 날짜)의 질문을 가져옵니다.

요청

GET /v1/questions/{qid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)

응답

200: 성공

```
{
  "id": "2022-01-01",
  "text": "산타를 몇살까지 믿었나요?",
  "answer_count": 0,
  "updated_at": "2022-04-18T18:51:32Z",
  "created_at": "2022-04-18T18:51:32Z"
}
```

이름	타입	설명
id	string	질문 아이디
text	string	질문
answer_count	integer	답 개수
updated_at	datetime	수정된 시간
created_at	datetime	생성된 시간

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

```
}
```

답 쓰기

답을 씁니다. 각 사용자는 하나의 질문에 하나의 답만 쓸 수 있습니다.

요청

POST /v1/questions/{qid}/answers

이름	위치	타입	필수	설명
qid	path	string	O	답을 쓸 질문 아이디 (yyyy-MM-dd)
text	body	string		답
photo	body	string		업로드 API 에서 받은 이미지 URL

응답

201: 성공

```
{
  "qid": "2022-04-18",
  "uid": "anonymous",
  "text": "5 살에 진실을 알았습니다.",
  "updated_at": "2022-06-02T13:10:26Z",
  "created_at": "2022-06-02T13:10:26Z"
}
```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	이미지 URL

updated_at	datetime	수정 시간
created_at	datetime	생성 시간

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

409: 이미 답이 있음

```
{
  "message": "Already answered"
}
```

답 가져오기

경로로 전달된 질문에 경로로 전달된 사용자가 쓴 답을 가져옵니다.

요청

GET /v1/questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디
uid	path	string	O	사용자 아이디

응답

200: 성공

```
{
  "qid": "2022-04-18",
  "uid": "anonymous",
```

```
{
  "text": "5 살에 진실을 알았습니다.",
  "updated_at": "2022-06-02T13:10:26Z",
  "created_at": "2022-06-02T13:10:26Z"
}
```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	이미지 URL
updated_at	datetime	수정된 시간
created_at	datetime	생성된 시간

404: 답을 찾을 수 없음

```
{
  "message": "Answer not found"
}
```

답 수정하기

내가 쓴 답을 수정합니다.

요청

PUT /v1/questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	수정할 질문 아이디 (yyyy-MM-dd)
uid	path	string	O	사용자 아이디

text	body	string	답
photo	body	string	업로드 API 에서 받은 이미지 URL

응답

200: 성공

```
{
  "qid": "2022-04-18",
  "uid": "anonymous",
  "text": "산타크로스는 진짜입니다.",
  "updated_at": "2022-06-02T13:52:04Z",
  "created_at": "2022-06-02T13:10:26Z"
}
```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	이미지 URL
updated_at	datetime	수정된 시간
created_at	datetime	생성된 시간

403: 권한 없음

```
{
  "message": "Permission denied"
}
```

404: 질문을 찾을 수 없음


```
{
  "message": "Question not found"
}
```

404: 답을 찾을 수 없음

```
{
  "message": "Answer not found"
}
```

답 삭제하기

내가 쓴 답을 삭제합니다.

요청

DELETE /v1/questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	삭제할 질문 아이디 (yyyy-MM-dd)
uid	path	string	O	사용자 아이디

응답

204: 성공

403: 권한 없음

```
{
  "message": "Permission denied"
}
```

404: 답을 찾을 수 없음

```
{
  "message": "Answer not found"
}
```

Version 2

Version 2 의 API 를 호출하기 위해서는 사용자 인증 정보를 포함해야 합니다. 인증 정보는 토큰 발급/갱신 API 에서 받을 수 있고, 여기서 받은 access_token 을 인증이 필요한 API 를 호출할 때 Authorization 헤더로 전달합니다.

```
Authorization: Bearer access_token
```

access_token 은 일정 시간이 지나면 사용할 수 없고, refresh_token 을 이용해 새로 발급 받아야 합니다.

Version 2 의 API 중 토큰 발급/갱신 API 는 Authorization 헤더를 필요로 하지 않습니다.

토큰 발급/갱신

아이디와 패스워드나 리프레시 토큰을 파라미터로 전달하거나 토큰을 발급받을 수 있습니다. grant_type 이 password 일 때 username 과 password 를, refresh_token 일 땐 refresh_token 을 사용합니다.

요청

POST /v2/token

이름	위치	타입	필수	설명
grant_type	body	string	O	승인 타입 - password - refresh_token
username	body	string		사용자 아이디

password	body	string	패스워드
refresh_token	body	string	리프레시 토큰

응답

204: 성공

```
{
  "token_type": "Bearer",
  "access_token": "eyJ0eXAiOiJK...99sPTAtFC1YFQ",
  "refresh_token": "eyJ0eXAiOiJK...g6TQ14ricfH0ws"
}
```

이름	타입	설명
token_type	string	토큰 타입
access_token	string	액세스 토큰 (5 분)
refresh_token	string	리프레시 토큰 (30 일)

400: 유효하지 않은 토큰

```
{
  "message": "Invalid refresh token"
}
```

400: 지원하지 않는 grant type

```
{
  "message": "Unsupported grant type"
}
```

401: 사용자 이름 또는 비밀번호가 정확하지 않음

```
{
```

```
"message": "The username or password was not correct"
}
```

푸시 토큰 등록

실시간 알림을 위한 FCM 등록 토큰을 서버에 저장합니다.

요청

POST /v2/user/push-tokens

이름	위치	타입	필수	설명
token	body	string	O	FCM 등록 토큰

응답

204: 성공

사용자 정보 가져오기

사용자의 프로필 정보를 가져옵니다. API 를 호출한 사용자의 팔로우 여부를 나타내는 is_following 필드가 추가됩니다.

요청

GET /v2/users/{uid}

이름	위치	타입	필수	설명
uid	path	string	O	조회할 사용자의 아이디

응답

200: 성공

```
{
```

```
{
  "id": "vincent",
  "name": "빈센트",
  "description": "나는 그림을 그리기 위해서 살아있다",
  "photo": "http://127.0.0.1:5000/v2/images/vincent.jpg",
  "answer_count": 604,
  "follower_count": 3,
  "following_count": 3,
  "is_following": false,
  "updated_at": "2022-06-02T08:38:00Z",
  "created_at": "2022-04-18T18:51:11Z"
}
```

이름	타입	설명
id	string	사용자 아이디
name	string	사용자 이름
description	string	자기소개
photo	string	프로필 이미지 URL
answer_count	integer	작성한 답 개수
follower_count	integer	팔로워 수
following_count	integer	팔로잉 수
is_following	boolean	나의 팔로우 여부
update_at	datetime	수정 시간
created_at	datetime	생성 시간

404: 사용자 정보를 찾을 수 없음

```
{
  "message": "User not found"
}
```

질문 가져오기

경로로 전달된 질문 아이디(특정 날짜)의 질문을 가져옵니다. 응답 헤더로 etag 를 내려줍니다.

다음 요청에서 If-None-Match 헤더로 전달받은 etag 를 전달해 변경 여부를 확인할 수 있습니다.

요청

GET /v2/questions/{qid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)
If-None-Match	header	string		응답 헤더의 etag

응답

200: 성공

```
etag: "c57e55dfe3e796cb6e423f28c7fe09ca"
```

```
{
  "id": "2022-04-18",
  "text": "주변의 녹색을 찾아보아요",
  "answer_count": 0,
  "updated_at": "2022-04-18T18:51:33Z",
  "created_at": "2022-04-18T18:51:33Z"
}
```

이름	타입	설명
id	string	질문 아이디
text	string	질문
answer_count	integer	답 개수
updated_at	datetime	수정 시간

created_at	datetime	생성 시간
------------	----------	-------

304: 변경 없음

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

질문 목록 가져오기

질문을 날짜의 역순으로 가져옵니다. from_date 부터 page_size 만큼의 질문을 가져옵니다.

예를들어 from_date 를 2022-01-05 로 요청하면 2022-01-05, 2022-01-04, 2022-01-03, 2022-01-02, 2022-01-01 다섯 개의 질문을 순서대로 가져옵니다.

요청

GET /v2/questions

이름	위치	타입	필수	설명
from_date	query	string	O	조회할 마지막 질문의 날짜 (yyyy-MM-dd)
page_size	query	integer		가져올 질문 개수 (기본 5 개)

응답

200: 성공

```
[
  {
    "id": "2022-04-18",
    "text": "주변의 녹색을 찾아보아요",
    "answer_count": 0,
    "updated_at": "2022-04-18T18:51:33Z",
    "created_at": "2022-04-18T18:51:33Z"
  }
]
```

```
},
"/* ... */",
{
  "id": "2022-04-14",
  "text": "내 묘비명은 무엇이 될까요?",
  "answer_count": 0,
  "updated_at": "2022-04-18T18:51:33Z",
  "created_at": "2022-04-18T18:51:33Z"
}
]
```

400: from_date 가 미래일 수 없음

```
{
  "message": "from_date is invalid"
}
```

사용자의 답 목록 가져오기

사용자가 작성한 답과 질문 목록을 질문 시간의 역순으로 가져옵니다.

요청

GET /v2/users/{uid}/answers

이름	위치	타입	필수	설명
uid	path	string	O	조회할 사용자의 아이디
from_date	query	string	O	조회할 질문의 마지막 날짜 (yyyy-MM-dd)
page_size	query	integer		가져올 질문 개수 (기본 3 개)

응답

200: 성공


```
[
  {
    "question": {
      "id": "2022-04-08",
      "text": "램프의 지니가 한 가지 소원을 들어준다면?",
      "answer_count": 2,
      "updated_at": "2022-04-18T18:51:33Z",
      "created_at": "2022-04-18T18:51:33Z"
    },
    "answer": {
      "qid": "2022-04-08",
      "uid": "vincent",
      "text": "가족과의 관계를 회복하고 싶어요",
      "updated_at": "2022-04-18T18:51:33Z",
      "created_at": "2022-04-18T18:51:33Z"
    }
  },
  /* ... */
  {
    "question": {
      "id": "2022-04-05",
      "created_at": "2022-04-18T18:51:33Z",
      "updated_at": "2022-04-18T18:51:33Z",
      "text": "혼자 시간을 보낼 때 주로 무엇을 하나요?",
      "answer_count": 2
    },
    "answer": {
      "created_at": "2022-04-18T18:51:33Z",
      "qid": "2022-04-05",
      "updated_at": "2022-04-18T18:51:33Z",
      "text": "그림을 그립니다. 항상 똑같죠.",
      "uid": "vincent"
    }
  }
]
```

이름

타입

설명

question	object	질문 오브젝트
id	string	질문 아이디
text	string	질문
answer_count	integer	답 개수
updated_at	datetime	수정 시간
created_at	datetime	생성 시간
answer	object	답 오브젝트
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	이미지 URL
updated_at	datetime	수정 시간
created_at	datetime	생성 시간

답 목록 가져오기

질문의 모든 답을 가져옵니다.

요청

GET /v2/questions/{qid}/answers

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)
If-None-Match	header	string	O	응답 헤더의 etag

응답

200: 성공

```
[
  {
    "qid": "2022-04-08",
    "uid": "vincent",
    "text": "가족과의 관계를 회복하고 싶어요",
    "answerer": {
      "id": "vincent",
      "name": "빈센트",
      "photo": "http://127.0.0.1:5000/v2/images/vincent.jpg",
      "description": "나는 그림을 그리기 위해서 살아있다",
      "answer_count": 604,
      "follower_count": 3,
      "following_count": 3,
      "updated_at": "2022-06-02T08:38:00Z",
      "created_at": "2022-04-18T18:51:11Z"
    },
    "updated_at": "2022-04-18T18:51:33Z",
    "created_at": "2022-04-18T18:51:33Z"
  },
  {
    "qid": "2022-04-08",
    "uid": "zuave",
    "text": "이번주 로또 당첨 번호를 알려줘",
    "answerer": {
      "id": "zuave",
      "name": "주아브 병사",
      "photo": "http://127.0.0.1:5000/v2/images/zuave.jpg",
      "description": "NO WAR",
      "answer_count": 400,
      "follower_count": 3,
      "following_count": 0,
      "updated_at": "2022-04-18T18:51:56Z",
      "created_at": "2022-04-18T18:51:11Z"
    },
    "updated_at": "2022-04-18T18:51:33Z",
    "created_at": "2022-04-18T18:51:33Z"
  }
]
```

]

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	질문
updated_at	datetime	수정 시간
created_at	datetime	생성 시간
answerer	object	답을 쓴 사용자 오브젝트
id	string	사용자 아이디
name	string	사용자 이름
description	string	자기소개
photo	string	프로필 이미지 URL
answer_count	integer	작성한 답 개수
follower_count	integer	팔로워 수
following_count	integer	팔로잉 수
update_at	datetime	수정 시간
created_at	datetime	생성 시간

답 쓰기

답을 씁니다. 각 사용자는 하나의 질문에 하나의 답만 쓸 수 있습니다.

요청

POST /v2/questions/{qid}/answers

이름	위치	타입	필수	설명
----	----	----	----	----

qid	path	string	O	질문 아이디 (yyyy-MM-dd)
text	body	string		답
photo	body	string		업로드 API 에서 받은 이미지 URL

응답

200: 성공

```
{
  "qid": "2021-04-01",
  "uid": "vincent",
  "text": "배가 고픈니다.",
  "updated_at": "2022-06-02T16:16:02Z",
  "created_at": "2022-06-02T16:16:02Z"
}
```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	답 이미지 URL
updated_at	datetime	수정 시간
created_at	datetime	생성 시간

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

409: 이미 답이 있음

```
{
  "message": "Already answered "
}
```

답 가져오기

경로로 전달된 질문에 경로로 전달된 사용자가 쓴 답을 가져옵니다. answerer 필드로 답을 쓴 사용자 정보를 가져옵니다.

요청

GET /v2/questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)
uid	path	string	O	사용자 아이디

응답

200: 성공

```
{
  "qid": "2022-04-08",
  "uid": "vincent",
  "text": "가족과의 관계를 회복하고 싶어요",
  "answerer": {
    "id": "vincent",
    "name": "빈센트",
    "photo": "http://127.0.0.1:5000/v2/images/vincent.jpg",
    "description": "나는 그림을 그리기 위해서 살아있다",
    "answer_count": 604,
    "follower_count": 3,
    "following_count": 3,
    "updated_at": "2022-06-02T08:38:00Z",
```

```

    "created_at": "2022-04-18T18:51:11Z"
  },
  "updated_at": "2022-04-18T18:51:33Z",
  "created_at": "2022-04-18T18:51:33Z"
}

```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	답 이미지 URL
updated_at	datetime	수정 시간
created_at	datetime	생성 시간
answerer	object	답을 쓴 사용자 오브젝트
id	string	사용자 아이디
name	string	사용자 이름
description	string	자기소개
photo	string	프로필 이미지 URL
answer_count	integer	작성한 답 개수
follower_count	integer	팔로워 수
following_count	integer	팔로잉 수
update_at	datetime	수정 시간
created_at	datetime	생성 시간

403: 권한 없음

```
{
  "message": "Permission denied"
}
```

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

404: 답을 찾을 수 없음

```
{
  "message": "Answer not found"
}
```

답 수정하기

내가 쓴 답을 수정합니다.

요청

PUT /questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)
uid	path	string	O	사용자 아이디
text	body	string		답
photo	body	string		업로드 API 에서 받은 이미지 URL

응답

200: 성공


```
{
  "qid": "2022-04-08",
  "uid": "vincent",
  "text": "가족과의 관계를 회복하고 싶어요",
  "updated_at": "2022-04-18T18:51:33Z",
  "created_at": "2022-04-18T18:51:33Z"
}
```

이름	타입	설명
qid	string	질문 아이디
uid	string	사용자 아이디
text	string	답
photo	string	답 이미지 URL
updated_at	datetime	수정 시간
created_at	datetime	생성 시간

403: 권한 없음

```
{
  "message": "Permission denied"
}
```

404: 질문을 찾을 수 없음

```
{
  "message": "Question not found"
}
```

404: 답을 찾을 수 없음

```
{
```

```
"message": "Answer not found"
}
```

답 삭제하기

내가 쓴 답을 삭제합니다.

요청

DELETE /v2/questions/{qid}/answers/{uid}

이름	위치	타입	필수	설명
qid	path	string	O	질문 아이디 (yyyy-MM-dd)
uid	path	string	O	사용자 아이디

응답

204: 성공

403: 권한 없음

```
{
  "message": "Permission denied"
}
```

404: 답을 찾을 수 없음

```
{
  "message": "Answer not found"
}
```

이미지 업로드

multipart/form-data 로 jpg 나 png 이미지를 업로드합니다. 응답으로 이미지의 URL 을 받습니다.

요청

POST /v2/images

이름	위치	타입	필수	설명
image	body	file	O	jpg/png 이미지 파일

응답

201: 성공

Location: http://127.0.0.1:5000/v2/images/bbbc44b1212d48c48f050c73353e09d8

```
{
  "url": "http://127.0.0.1:5000/v2/images/91c62eed7c3d4e7b9a82783589857ec8"
}
```

이름	타입	설명
url	string	업로드한 이미지의 URL

팔로우

다른 사용자를 팔로우합니다. 팔로우 대상에게 실시간 알림이 전송되고, 팔로우 대상이 새로운 답을 썼을 때 나에게 실시간 알림이 전송됩니다.

요청

POST /v2/user/following/{followee_uid}

이름	위치	타입	필수	설명
followee_uid	path	string	O	팔로우 할 대상의 사용자 아이디

응답

204: 성공

400: 자기를 팔로우 할 수 없음

```
{
  "message": "Can't follow yourself"
}
```

404: 사용자를 찾을 수 없음

```
{
  "message": "User not found"
}
```

409: 이미 팔로우 하고 있음

```
{
  "message": "Following already"
}
```

팔로우 취소

팔로우를 취소합니다.

요청

DELETE /v2/user/following/{followee_uid}

이름	위치	타입	필수	설명
followee_uid	path	string	O	팔로우 한 사용자 아이디

응답

204: 성공

400: 자기를 팔로우 할 수 없음

```
{  
  "message": "Can't follow yourself"  
}
```

404: 팔로우 하고 있지 않음

```
{  
  "message": "Not following "  
}
```

공통 오류

모든 API 에서 발생할 수 있는 공통 오류입니다.

401: 인증 토큰 없음

```
{  
  "msg": "Missing Authorization Header"  
}
```

401: 인증 토큰 만료

```
{  
  "msg": "Token has expired"  
}
```

422: JWT 토큰 오류

```
{  
  "message": "Not enough segments"  
}
```

```
{  
  "message": "Signature verification failed"  
}
```

422: 잘못된 파라미터

```
{
  "message": "The request was well-formed but was unable to be followed due to semantic errors."
}
```

알림 메시지

FCM 을 통해 전송되는 실시간 메시지 구조입니다.

팔로우 알림

이름	타입	설명
type	string	follow
uid	string	팔로어의 아이디
username	string	팔로어의 이름

새 답 알림

이름	타입	설명
type	string	answer
qid	string	질문 아이디
uid	string	답을 쓴 사람의 아이디
username	string	답을 쓴 사람의 이름

테스트 계정

이름	아이디	패스워드	비고
빈센트	vincent	12345678	
롤랑	postman	12345678	
탕기	tanguy	12345678	
세가토리	segatori	12345678	
주아브 병사	zuave	12345678	
가세 박사	ginoux	12345678	
지누 부인	gachet	12345678	
아무개	anonymous	12345678	V1 에서 인증 정보 없을 때 사용