

API 명세서 - 3. 게시글 API

전체 API 목록

API	분류	Method	API 명	접근 권한	설명
게시글	게시글 조회	GET	/api/article/all	permitAll	게시글 전체 조회
		GET	/api/article/all/user/{userId}	permitAll	특정 유저의 게시글 전체 조회
		GET	/api/article/{articleId}/user/{userId}	permitAll	특정 유저의 게시글 단건 조회
	게시글 등록	POST	/api/article/save/user/{userId}	authenticated	특정 유저의 게시글 등록
	게시글 수정	PUT	/api/article/{articleId}/user/{userId}	authenticated	특정 유저의 게시글 수정
	게시글 삭제	POST	/api/article/delete/user/{userId}	authenticated	특정 유저의 게시글 여러건 삭제
		POST	/api/article/delete/withdrawal	permitAll	탈퇴 유저의 게시글 삭제

게시글 조회 API

단건 조회

Request

Request Syntax

```
1 curl -X GET 'http://localhost/api/article/{articleId}/user/{userId}'
```

API 호출 방식

메서드	요청 url
GET	/api/article/{articleId}/user/{userId}

- path parameter : article id, user id

Response

게시글 조회 성공 응답 예시

```
1 {
2   "status": true,
3   "message": "article 1 조회",
4   "result": {
5     "id": 1,
6     "articleId": 1,
7     "title": "게시글 1번입니다.",
8     "content": "안녕하세요 게시글 1번 본문입니다.",
9     "modified_at": "2023-12-10T21:54:32.321",
10    "created_at": "2023-12-10T21:54:32.321",
11    "user": {
12      "id": 1,
13      "email": "user@email.com",
14      "nickname": "닉네임"
15    }
16  }
17 }
```

실패 응답 예시 [↗](#)

```
1 {
2   "status": false,
3   "message": "잘못된 게시글 번호입니다."
4 }
```

전체 조회 [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X GET 'http://localhost/api/article/all'
```

API 호출 방식 [↗](#)

메서드	요청 url
GET	/api/article/all

Response [↗](#)

게시글 조회 성공 응답 예시 [↗](#)

```
1 {
2   "status": true,
3   "message": "article 전체 조회",
4   "result": [
5     {
6       "id": 1,
```

```

7      "articleId": 1,
8      "title": "게시글 1번입니다.",
9      "content": "안녕하세요 게시글 1번 본문입니다.",
10     "modified_at": "2023-12-10T21:54:32.321",
11     "created_at": "2023-12-10T21:54:32.321",
12     "user": {
13         "id": 1,
14         "email": "user@email.com",
15         "nickname": "ㄱ"
16     }
17 },
18 {
19     "id": 2,
20     "articleId": 1,
21     "title": "1번입니다.",
22     "content": "안녕하세요 게시글 1번 본문입니다.",
23     "modified_at": "2023-12-10T21:54:32.321",
24     "created_at": "2023-12-10T21:54:32.321",
25     "user": {
26         "id": 2,
27         "email": "user@email.com",
28         "nickname": "ㄴ"
29     }
30 },
31 {
32     "id": 3,
33     "articleId": 1,
34     "title": "게시글 1번.",
35     "content": "안녕하세요 게시글 1번 본문입니다.",
36     "modified_at": "2023-12-10T21:54:32.321",
37     "created_at": "2023-12-10T21:54:32.321",
38     "user": {
39         "id": 3,
40         "email": "user@email.com",
41         "nickname": "ㄷ"
42     }
43 }
44 }

```

특정 유저 게시글 전체 조회 [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X GET 'http://localhost/api/article/all/user/{userId}'
```

API 호출 방식 [↗](#)

메서드	요청 url
GET	/api/article/all/user/{userId}

- path parameter : user id

Response

게시글 조회 성공 응답 예시

```
1 {
2   "status": true,
3   "message": "닉네임 article 전체 조회",
4   "result": [
5     {
6       "id": 1,
7       "articleId": 1,
8       "title": "게시글 1번입니다.",
9       "content": "안녕하세요 게시글 1번 본문입니다.",
10      "modified_at": "2023-12-10T21:54:32.321",
11      "created_at": "2023-12-10T21:54:32.321",
12      "user": {
13        "id": 1,
14        "email": "user@email.com",
15        "nickname": "닉네임"
16      }
17    },
18    {
19      "id": 2,
20      "articleId": 2,
21      "title": "2번입니다.",
22      "content": "안녕하세요 게시글 2번 본문입니다.",
23      "modified_at": "2023-12-10T21:54:32.321",
24      "created_at": "2023-12-10T21:54:32.321",
25      "user": {
26        "id": 1,
27        "email": "user@email.com",
28        "nickname": "닉네임"
29      }
30    },
31    {
32      "id": 3,
33      "articleId": 3,
34      "title": "게시글 3번.",
35      "content": "안녕하세요 게시글 3번 본문입니다.",
36      "modified_at": "2023-12-10T21:54:32.321",
37      "created_at": "2023-12-10T21:54:32.321",
38      "user": {
39        "id": 1,
40        "email": "user@email.com",
41        "nickname": "닉네임"
42      }
43    }
44  ]
45 }
```

게시글 등록 API [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X POST 'http://localhost/api/article/save/user/{userId}'
```

API 호출 방식 [↗](#)

메서드	요청 url
POST	/api/article/save/user/{userId}

- path parameter : user id

Request Body [↗](#)

```
1 {
2     "user_id": 1,
3     "title": "제목",
4     "content": "내용"
5 }
```

파라미터	타입	필수 여부	설명
useR_id	Long	필수	사용자 번호
title	String	필수	제목
content	String	필수	본문

Response [↗](#)

Response Header [↗](#)

```
1 HTTP/1.1 201 Created
2 Location: http://localhost:8080/api/article/all/user/{userId}
```

Response Body [↗](#)

게시글 등록 성공 응답 예시 [↗](#)

```
1 {
2     "status": true,
3     "message": "등록 되었습니다."
4 }
```

실패 응답 예시 - 제목이 없는 경우 [↗](#)

```
1 {
2   "status": false,
3   "message": "제목을 입력해주세요."
4 }
```

- 제목, 본문에 값이 없는 경우 실패 응답

게시글 수정 API [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X PUT 'http://localhost/api/article/{articleId}/user/{userId}'
```

API 호출 방식 [↗](#)

메서드	요청 url
PUT	/api/article/{articleId}/user/{userId}

- path parameter : user id, article id

Request Body [↗](#)

```
1 {
2   "articleId": 1,
3   "userId": 1,
4   "title": "",
5   "content": ""
6 }
```

파라미터	타입	필수 여부	설명
userId	Long	필수	유저 id
articleId	Long	필수	수정할 article id
title	String	필수	제목
content	String	필수	본문

Response [↗](#)

Response Header [↗](#)

```
1 HTTP/1.1 201 Created
2 Location: http://localhost:8080/api/article/{articleId}/user/{userId}
```

Response Body [↗](#)

게시글 등록 성공 응답 예시 [↗](#)

```
1 {
2   "status": true,
3   "message": "수정 되었습니다."
4 }
```

실패 응답 예시 - 제목이 없는 경우 [↗](#)

```
1 {
2   "status": false,
3   "message": "제목을 입력해주세요."
4 }
```

- 제목, 본문에 값이 없는 경우 실패 응답

실패 응답 예시 - 내가 작성하지 않은 게시글의 수정 요청 [↗](#)

```
1 {
2   "status": false,
3   "message": "잘못된 요청입니다."
4 }
```

실패 응답 예시 - 없는 게시글 id에 대한 수정 요청 [↗](#)

```
1 {
2   "status": false,
3   "message": "잘못된 요청입니다."
4 }
```

게시글 삭제 API [↗](#)

게시글 삭제 [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X POST 'http://localhost/api/article/delete/user/{userId}'
```

API 호출 방식 [↗](#)

메서드	요청 url
POST	/api/article/delete/user/{userId}

Request Body [↗](#)

```
1 {
2   "userId": 1,
3   "articleId": [1,2,3]
4 }
```

파라미터	타입	필수 여부	설명
userId	Long	필수	유저 id
articleId	List<Long>	필수	삭제할 article id 번호 리스트

Response [↗](#)

성공 응답 [↗](#)

Response Header [↗](#)

```
1 HTTP/1.1 204 No content
2 Location: http://localhost:8080/{userId}/articles
```

실패 응답 - 내가 작성하지 않은 게시글의 삭제 요청 [↗](#)

Response Header [↗](#)

```
1 HTTP/1.1 404 Bad Request
```

```
1 {
2   "status": false,
3   "message": "잘못된 요청입니다."
4 }
```

탈퇴자 게시글 삭제 [↗](#)

Request [↗](#)

Request Syntax [↗](#)

```
1 curl -X POST 'http://localhost/api/article/delete/withdrawal'
```


API 호출 방식 [↗](#)

메서드	요청 url
POST	/api/article/delete/withdrawal

Request Body [↗](#)

1	[1, 2, 3]		
파라미터	타입	필수 여부	설명
user id	List<Long>	필수	삭제할 user id 번호 리스트

Response [↗](#)

성공 응답 [↗](#)

Response Header [↗](#)

1	HTTP/1.1 204 No content
2	Location: http://localhost:8080/{userId}/articles

실패 응답 - 탈퇴하지 않은 사용자에게 대한 삭제 요청 시 [↗](#)

Response Header [↗](#)

1	HTTP/1.1 400 Bad Request
1	{
2	"status": false,
3	"message": "- user id : [1] email : [gimwlgus@kakao.com] 탈퇴 회원이 아닙니다.\n"
4	}

