

# API 명세서 - 3. 게시글 API

## 전체 API 목록

API	분류	Method	API 명	설명
게시글	게시글 조회	GET	/articles	게시글 전체 조회
		GET	/articles/{id}	게시글 단건 조회
	게시글 등록	POST	/article	게시글 등록
	게시글 수정	PUT	/articles/{id}	게시글 수정
	게시글 삭제	DELETE	/articles	게시글 여러건 삭제
		DELETE	/articles/{id}	게시글 단건 삭제

## 게시글 조회 API

### 단건 조회

#### Request

##### Request Syntax

```
1 curl -X GET 'http://localhost/articles/{id}'
```

#### API 호출 방식

메서드	요청 url
GET	/articles/{id}

- path parameter : article id

#### Response

##### 게시글 조회 성공 응답 예시

```
1 {
2   "status": true,
3   "message": "article 1 조회",
4   "result": {
5     "id": 1,
6     "title": "게시글 1번입니다.",
7     "content": "안녕하세요 게시글 1번 본문입니다.",
8     "modified_at": "2023-12-10T21:54:32.321",
9     "created_at": "2023-12-10T21:54:32.321"
```

```
10 }
11 }
```

#### 실패 응답 예시 [↗](#)

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

#### 전체 조회 [↗](#)

#### Request [↗](#)

##### Request Syntax [↗](#)

```
1 curl -X GET 'http://localhost/articles'
```

#### API 호출 방식 [↗](#)

메서드	요청 url
GET	/articles

#### Response [↗](#)

##### 게시글 조회 성공 응답 예시 [↗](#)

```
1 {
2   "status": true,
3   "message": "article 전체 조회",
4   "result": [
5     {
6       "id": 1,
7       "title": "게시글 1번입니다.",
8       "content": "안녕하세요 게시물 1번 본문입니다.",
9       "modified_at": "2023-12-10T21:54:32.321",
10      "created_at": "2023-12-10T21:54:32.321"
11    },
12    {
13      "id": 2,
14      "title": "2번입니다.",
15      "content": "안녕하세요 게시물 2번 본문입니다.",
16      "modified_at": "2023-12-10T21:54:32.321",
17      "created_at": "2023-12-10T21:54:32.321"
18    },
19    {
20      "id": 3,
21      "title": "게시글 3번.",
```

```
22         "content": "안녕하세요 게시글 3번 본문입니다.",
23         "modified_at": "2023-12-10T21:54:32.321",
24         "created_at": "2023-12-10T21:54:32.321"
25     }
26 }
```

## 게시글 등록 API [🔗](#)

### Request [🔗](#)

#### Request Syntax [🔗](#)

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

#### API 호출 방식 [🔗](#)

메서드	요청 url
POST	/article

#### 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/articles/{id}
```

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/articles/{id}'
```

API 호출 방식 [↗](#)

메서드	요청 url
PUT	/articles/{id}

- Path Parameter : article id

Request Body [↗](#)

```
1 {
2   "id": 1,
3   "title": "",
4   "content": ""
5 }
```

파라미터	타입	필수 여부	설명
id	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/articles/{id}
```

### 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 1개 삭제 [↗](#)

#### Request [↗](#)

##### Request Syntax [↗](#)

```
1 curl -X DELETE 'http://localhost/articles/{id}'
```

#### API 호출 방식 [↗](#)

메서드	요청 url
DELETE	/articles/{id}

- Path Parameter : article id

#### Response [↗](#)

#### 성공 응답 [↗](#)

##### Response Header [↗](#)

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

## 게시글 여러개 삭제 [↗](#)

#### Request [↗](#)

##### Request Syntax [↗](#)

```
1 curl -X DELETE 'http://localhost/articles'
```

API 호출 방식 [↗](#)

메서드	요청 url
DELETE	/articles

Request Body [↗](#)

```
1 {
2   "id": [1,2,3]
3 }
```

파라미터	타입	필수 여부	설명
id	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/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 }
```

