

게시판 관리 API 명세서

1. GET - 전체 게시물 목록 조회 API

- 설명
 - 제목, 작성자명, 내용, 작성 날짜가 포함된 모든 게시물 목록 검색
 - 작성 날짜 기준으로 내림차순 정렬
- EndPoint
 - GET /api/boards
- Response

```
{
  "status": 200,
  "success": true,
  "message": "전체 게시물 조회에 성공했습니다.",
  "data": {
    [
      {
        "title": "Post 1",
        "username": "John Doe",
        "content": "This is the content of post 1.",
        "createdAt": "2024-11-10T10:00:00",
        "updatedAt": "2024-11-10T10:00:00"
      },
      {
        "title": "Post 2",
        "username": "Jane Doe",
        "content": "This is the content of post 2.",
        "createdAt": "2024-11-09T09:00:00",
        "updatedAt": "2024-11-09T09:00:00"
      }
    ]
  }
}
```

```
}  
]  
}
```

2. POST - 게시글 작성 API

- 설명
 - 제목, 작성자명, 비밀번호, 내용을 입력하여 새 게시물 저장 후 게시물 반환
- EndPoint
 - POST /api/board

- Request

```
{  
  "title": "New Post",  
  "content": "This is a new post content."  
}
```

- Response

```
{  
  "status": 200,  
  "success": true,  
  "message": "게시물 등록에 성공했습니다.",  
  "data": {  
    "title": "New Post",  
    "username": "John Doe",  
    "content": "This is a new post content.",  
    "createdAt": "2024-11-10T12:00:00",  
    "updatedAt": "2024-11-10T10:00:00"  
  }  
}
```

3. GET - 선택한 게시글 조회 API

- 설명
 - 게시글 ID로 단일 게시물의 제목, 작성자명, 생성일, 내용 조회
- EndPoint
 - GET /api/board/{id}
- Response

```
{
  "status": 200,
  "success": true,
  "message": "게시물 조회에 성공했습니다.",
  "data": {
    "title": "Post 1",
    "username": "John Doe",
    "content": "This is the content of post 1.",
    "createdAt": "2024-11-10T10:00:00",
    "updatedAt": "2024-11-10T10:00:00"
  }
}
```

4. PUT - 선택한 게시글 수정 API

- 설명
 - 제공된 비밀번호 검증 후 게시물 제목, 작성자명, 작성 내용 수정
 - 수정 후 게시글 정보 반환
- EndPoint
 - PUT /api/board/{id}
- Request

```
{
  "title": "Updated Title",
  "content": "Updated content for the post."
}
```

- Response

```
{
  "status": 200,
  "success": true,
  "message": "게시물 조회에 성공했습니다.",
  "data": {
    "title": "Updated Title",
    "username": "John Doe",
    "content": "Updated content for the post.",
    "createdAt": "2024-11-10T10:00:00",
    "updatedAt": "2024-11-10T10:00:00"
  }
}
```

5. DELETE - 선택한 게시글 삭제 API

- 설명
 - 삭제하기 전에 비밀번호를 검증한 후 게시물 삭제
 - 삭제 후 성공 메시지 반환
- EndPoint
 - DELETE /api/board/{id}
- Response

```
{
  "status": 200,
  "success": true,
  "message": "게시물 삭제에 성공했습니다."
}
```

6. POST - 회원가입 API

- 설명
 - username, password 전달 받아서 저장
 - 중복된 username 없으면 회원 저장 후 성공 메시지, 상태코드 반환
- EndPoint
 - POST /api/auth/signup
- Request

```
{
  "username": "John Doe"
  "password": "password123"
}
```

- Response

```
{
  "status": 200,
  "success": true,
  "message": "회원 가입에 성공했습니다."
}
```

7. POST - 로그인 API

- 설명
 - username, password 전달 받아서 로그인 진행
- EndPoint
 - POST /api/auth/signin
- Request

```
{  
  "username": "John Doe"  
  "password": "password123"  
}
```

- Response

```
{  
  "status": 200,  
  "success": true,  
  "message": "로그인에 성공했습니다."  
}
```