날:숨 REST API 연동규격서

**실어증 환자 치료 플랫폼 “날:숨” REST API 연동 규격서**

**개정 이력**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **변경일자** | **계정 내용** | **작성자** |
| 0.0.1 | 2024/01/12 | 초안 작성 | 김희중 |
| 0.0.2 | 2024/1/19 | Board API 파라미터 정보 수정 | 김희중 |
| 0.0.3 | 2024/1/26 | 전체 API 수정 - common API 내용 추가  - user API 내용 수정  - rahbilitation API 내용 수정 및 추가  multipart/form-data 정보 추가 | 김희중 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**목 차**

**1. 개요**

1.1 목적

1.2 용어 정의, 약자, 약어

1.3 참고 문헌

**2. API 연동 규격**

**2.1 회원 기능**

2.1.1 플랫폼 회원가입

2.1.2 플랫폼 로그인

2.1.3 아이디 중복 확인 요청

2.1.4 이메일 중복 확인 요청

2.1.5 이메일 인증코드 생성 요창

2.1.6 이메일 인증 요청(인증코드 확인 요청)

2.1.7닉네임 재설정

2.1.8 프로필 사진 재설정

2.1.9 임시 비밀번호 발급

2.1.10 비밀번호 인증 요청

2.1.11 비밀번호 변경 요청

2.1.12카카오 회원가입

2.1.13카카오 로그인

2.1.14 Refresh Token 재발급

**2.2 재활 코스 문제 서비스**

2.2.1 재활 코스 목록 조회

2.2.2 재활 코스 카테고리 조회

2.2.3 재활 치료 문제집 요청

2.2.4 문제 정답/오답 저장

2.2.5 유창성 문제 정답 처리

2.2.6 복습 문제 저장

2.2.7 복습 문제집 요청

**2.3 게시판 서비스**

2.3.1 게시판 분류 목록 조회

2.3.2 게시판 글 목록 조회

2.3.3 게시글 상세 조회

2.3.4 게시글 생성

2.3.5 게시글 수정

2.3.6 게시글 삭제

2.3.7 게시글 검색

2.3.8 이미지 저장

2.3.9 이미지 삭제

2.3.10 게시글 댓글 조회

2.3.11 댓글 작성

2.3.12 댓글 수정

2.3.13 댓글 삭제

**1. 개요**

**1.1 목적**

이 문서는 Web Client와 Application Server 간의 연동 통신 규격을 명시하는 문서이다.

**1.2 용어 정의, 약자, 약어**

|  |  |
| --- | --- |
| 용 어 | 내 용 |
| HTTP | Hypertext Transfer Protocol. |
| HTTPS | Hypertext Transfer Protocol over Secure Socket Layer. |
| JSON | Java Script Object Notation |
| multipart/from-data | HTTP 요청 또는 응답에서 여러 종류의 데이터를 동시에 전송하기 위해 사용되는 방식. 파일 전송을 위해 사용되는 형식. |

* 1. **참고 문서**
* 없음

**2. API 연동 규격**

**2.1 회원 기능**

**2.1.1 플랫폼 회원가입**

사용자 id, 이름, 닉네임, 이메일, 비밀번호, 생년월일 정보를 받아 사용자 신규 생성시 사용

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/join
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| login\_id | Int | **O** | 사용자 가입 아이디 |  |
| password | String | **O** | 사용자 가입 비밀번호 |  |
| name | String | **O** | 사용자 이름 |  |
| nickname | String | **O** | 사용자 닉네임 |  |
| email | String | **O** | 사용자 이메일 ID |  |
| email\_domain | String | **O** | 사용자 이메일 도메인 |  |
| birth | String | **O** | 사용자 생년월일 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| - | - | **-** | - | 200 : 회원가입 성공 400 : 회원가입 실패 |

**2.1.2 플랫폼 로그인**

날숨 플랫폼에서 사용자 id, 비밀번호를 입력받아 사용자 정보를 확인하고 토큰을 발급한다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/login
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| login\_id | Int | **O** | 사용자 가입 아이디 |  |
| password | String | **O** | 사용자 가입 비밀번호 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| token | 1 | 해당 유저의 token값 | String | access token과 refresh token 발급  토큰에는 member\_id와 role 정보가 들어간다 |

**2.1.3 아이디 중복확인 요청**

사용자 입력 id가 이미 존재하는 id인지 확인한다. status 코드에 따라 응답 결과를 확인한다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/id
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| login\_id | Int | **O** | 사용자 기입 아이디 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 2 : 이미 존재하는 id 3 : id 규칙 위반 4 : id 데이터 부재 |

**2.1.4 이메일 중복확인 요청**

사용자 입력 email이 이미 존재하는 email인지 확인한다. status 코드에 따라 응답 결과를 확인한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/email
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| email\_id | String | **O** | 이메일 아이디 | 이메일 앞부분 |
| email\_domain | String | **O** | 이메일 도메인 | 이메일 뒷부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 2 : 이미 존재하는 email 3 : email 규칙 위반 4 : email 데이터 부재 |

**2.1.5 이메일 인증코드 생성 요청**

인증번호를 생성하여 사용자 이메일로 보낸다. **2.1.6 이메일 인증 요청**을 통한 이메일 인증을 위해 사용된다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/email/request
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| email\_id | String | **O** | 이메일 아이디 | 이메일 앞부분 |
| email\_domain | String | **O** | 이메일 도메인 | 이메일 뒷부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 2 : 이미 존재하는 email 3 : email 규칙 위반 4 : email 데이터 부재 |
| code | 1 | 인증코드 | String | email 인증을 위한 코드. |

**2.1.6 이메일 인증 요청(인증코드 확인 요청)**

이메일 인증을 위해 사용자가 입력한 코드가 올바른 인증코드인지 확인한다. **2.1.5 이메일 인증코드 생성 요청**을 통해 코드를 생성받는다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/email/check
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| code | String | **O** | 인증코드 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 2 : 이미 존재하는 email 3 : email 형식 규칙 위반 4 : email 데이터 부재 |

**2.1.7 닉네임 재설정**

사용자 닉네임을 재설정한다. access token 정보를 통해 사용자를 확인한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/nickname
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰값 | header로 전달 |
| nickname | String | **O** | 사용자 입력 nickname |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 4 : 데이터 부재 |

**2.1.8 프로필 사진 재설정**

사용자 닉네임을 재설정한다. access token 정보를 통해 사용자를 확인한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/nickname
* Header Content-Type = multipart/form-data
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰값 | header로 전달 |
| image | file | **O** | 사용자 프로필 이미지 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 4 : 데이터 부재 |

**2.1.9 임시 비밀번호 발급**

비밀번호 분실시 임시 비밀버호 재발급을 요청한다. id, email 정보 인증 과정을 거친 후 임시 비밀번호를 사용자 email로 보낸다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/temp-password
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| login\_id | String | **O** | 사용자 로그인 ID |  |
| email\_id | String | **O** | 이메일 아이디 | 이메일 앞부분 |
| email\_domain | String | **O** | 이메일 도메인 | 이메일 뒷부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 유효성 검증 실패 4 : 데이터 부재 |

**2.1.10 비밀번호 인증 요청**

사용자가 입력한 비밀번호 인증을 요청한다. token값 정보를 통해 비밀번호를 비교하여 결과 status를 반환한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/check-password
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰값 | header로 전달 |
| password | String | **O** | 사용자 입력 비밀번호 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 비밀번호 형식 위반 4 : 비밀번호 데이터 부재 |

**2.1.11 비밀번호 변경 요청**

사용자의 기존 비밀번호를 확인하고 사용자가 입력한 새로운 비밀번호로 사용자 비밀번호를 변경한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/repassword
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰값 | header로 전달 |
| new\_password | String | **O** | 사용자 기존 비밀번호 |  |
| old\_pawword | String | **O** | 사용자 신규 비밀번호 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 비밀번호 형식 위반 4 : 비밀번호 데이터 부재 |

**2.1.12 카카오 회원가입**

카카오 소셜 로그인을 통한 회원가입을 진행한다. 카카오 소셜 로그인을 통해 받은 code를 입력하면 서버측에서 회원가입을 진행시켜준다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/outh/kakao/login
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| code | String | **O** | 카카오 인증 코드 | 앞부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 2 : 이미 존재하는 id 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.1.13 카카오 로그인**

카카오 소셜 로그인을 통한 플랫폼 로그인을 진행한다. 카카오 소셜 로그인을 통해 받은 code를 입력하면 서버측에서 토큰을 발급해준다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/outh/kakao/login
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| code | String | **O** | 카카오 인증 코드 | 앞부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| token | 1 | 해당 유저의 token값 | String | access token과 refresh token 발급  토큰에는 member\_id와 role 정보가 들어간다 |

**2.1.14 Refresh Token 재발급**

refresh token 재발급을 요청한다.

.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/refresh
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| code | String | **O** | 카카오 인증 코드 | 앞부분 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| token | 1 | 해당 유저의 token값 | String | access token과 refresh token 발급  토큰에는 member\_id와 role 정보가 들어간다 |

**2.2 재활 코스 문제 서비스**

**2.2.1 재활 코스 목록 조회**

재활 코스 목록(이름 대기, 따라 말하기, 듣기 이해력, 유창성)을 조회한다..

* **Request**
* Method = GET 방식
* HTTP URL = ‘/api/rehabilitation
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| - | - | **-** | - |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| course\_id | 2 | 재활 치료 코스 id | Int |  |
| course\_name | 2 | 재활 치료 코스 이름 | String |  |

**2.2.2 재활 코스 카테고리 조회**

재활 치료 코스별 카테고리 정보를 조회한다..

* **Request**
* Method = GET 방식
* HTTP URL = ‘/api/rehabilitation/{course\_id}’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| course\_id | int | **O** | 재활 치료 코스 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| category\_id | 2 | 재활 치료 코스별 문제 정보 id | Int |  |
| category\_name | 2 | 재활 치료 코스 이름 | String |  |
| over\_time\_sec | 2 | 문제 제한시간 | int |  |

* + 1. **재활치료 문제집 요청**

문제 풀이를 위한 재활 치료 문제집(문제 모음)을 요청한다. category id를 통해 course id 조회 후 course id에 알맞은 response 데이터를 제공한다.  
 - course\_id 번호

1. : 이름 대기
2. : 따라 말하기
3. : 듣기 이해력(그림 짯짓기)
4. : 듣기 이해력(설명 듣고 올바른 텍스트 고르기)
5. : 유창성

* **Request**
* Method = GET 방식
* HTTP URL = ‘/api/rehabilitation/problem/{category\_id}’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 token 정보 | 플랫폼 사용자만 문제 풀이 이용 가능 |
| category\_id | Int | **O** | 재활 치료 코스별 카테고리 id | category id를 통해 문제 course 및 category 모두 조회 가능 |

* **Response**
* 반환 타입 = **JSON 형태**
* **이름 대기 Response [course\_id = 0]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 이름 대기 문제 id | Int |  |
| image | 2 | 이름 대기 문제 이미지 | String |  |
| answer | 2 | 문제 정답 | String |  |
| hint | 2 | 힌트 | String |  |

* **따라 말하기 Response [course \_id = 1]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 따라 말하기 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| hint | 2 | 문제 힌트 | String |  |

* **듣기 이해력(그림 짯짓기) Response [course \_id = 2]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 듣기 이해력(그림 짝짓기) 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| options | 2 | 문제 선택지 (이미지 리스트) | String | 4개의 이미지 URL 주어짐 |
| answer | 2 | 정답 번호 | Int | 정답 선택지 번호 |

* **듣기 이해력(텍스트 고르기) Response [course \_id = 3]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 듣기 이해력(그림 짝짓기) 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| question\_image | 2 | 문제 이미지 | String |  |
| options | 2 | 문제 선택지 (Text) | String | 4개의 텍스트 선택지 주어짐 |
| answer | 2 | 정답 번호 | String | 정답 선택지 번호 |

* **유창성 문제 Response [course \_id = 4]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 재활 치료 코스별 문제 정보 id | Int |  |
| question | 2 | 문제 내용 | String |  |

**2.2.4 문제 정답/오답 저장**

유창성을 제외한 문제의 정답/오답 여부를 저장한다. 정답 여부는 클라이언트 측에서 문제와 함께 주었던 정답지를 통해 확인한 후 정답 여부를 서버로 보내준다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/rehabilitation/result
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| course\_id | int | **O** | 재활 치료 코스 정보 id |  |
| problem\_id | int | **O** | 문제 id |  |
| is\_right | boolean | **O** | 해당 문제 정답 여부 ( 정답 Y / 오답 N ) |  |
| time | int/null | **O** | 해당 문제를 푼 시간 | 초 단위로 제공 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** |  |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.2.5 유창성 문제 정답 처리**

유창성 문제 답변에 대한 정답 여부를 처리한다. 유창성 문제에 대한 사용자 답변의 적절성 여부를 AI 모델을 통해 판단 및 설명하며 도출된 데이터를 반환한다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/rehabilitation/fluency-check
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| question | String | **O** | 사용자가 푼 유창성 문제 정보 |  |
| answer | String | **O** | 사용자가 제시한 답변 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| result | 1 | 사용자 문제에 대한 결과 | boolean |  |
| explain | 1 | 결과 설명 |  |  |

**2.2.6 복습 문제 저장**

복습하고자 하는 문제를 복습 문제로 저장한다. 문제 정보 데이터를 받아 사용자의 복습 문제 리스트에 저장한다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/rehabilitation/review
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| course\_id | int | **O** | 재활 치료 코스 정보 id |  |
| problem\_id | int | **O** | 문제 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** |  |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.2.7 복습 문제집 요청**

복습용 문제로 저장한 문제들의 문제집(문제 목록)을 요청한다. couse\_id를 통해 복습하고자 하는 코스 문제집을 요청한다.

* **Request**
* Method = POST 방식
* HTTP URL = ‘/api/rehabilitation/review/{couse\_id}
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| course\_id | int | **O** | course\_id |  |

* **Response**
* 반환 타입 = **JSON 형태**
* **이름 대기 Response [course\_id = 0]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 이름 대기 문제 id | Int |  |
| image | 2 | 이름 대기 문제 이미지 | String |  |
| answer | 2 | 문제 정답 | String |  |
| hint | 2 | 힌트 | String |  |

* **따라 말하기 Response [course \_id = 1]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 따라 말하기 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| hint | 2 | 문제 힌트 | String |  |

* **듣기 이해력(그림 짯짓기) Response [course \_id = 2]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 듣기 이해력(그림 짝짓기) 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| options | 2 | 문제 선택지 (이미지 리스트) | String | 4개의 이미지 URL 주어짐 |
| answer | 2 | 정답 번호 | Int | 정답 선택지 번호 |

* **듣기 이해력(텍스트 고르기) Response [course \_id = 3]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 듣기 이해력(그림 짝짓기) 문제 id | Int |  |
| question | 2 | 문제 내용 | String |  |
| question\_image | 2 | 문제 이미지 | String |  |
| options | 2 | 문제 선택지 (Text) | String | 4개의 텍스트 선택지 주어짐 |
| answer | 2 | 정답 번호 | String | 정답 선택지 번호 |

* **유창성 문제 Response [course \_id = 4]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| problem\_id | 2 | 재활 치료 코스별 문제 정보 id | Int |  |
| question | 2 | 문제 내용 | String |  |

**2.3 게시판 서비스**

**2.3.1 게시판 분류 목록 조회**

게시판 분류 목록(게시판 카테고리)을 확인 할 수 있다.

* **Request**
* Method = GET방식
* HTTP URL = ‘/api/boards
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| - | - | **-** | - |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| category\_id | 1 | 게시판 카테고리 id | Int |  |
| category\_name | 1 | 게시판 목록 이름 | String |  |

**2.3.2 게시판 글 목록 조회**

게시판 분류 id, 정렬 방식을 통해 글 목록을 가져온다. 페이지 번호 입력을 통해 받아오는 글의 수를 조절한다..

* **Request**
* Method = GET방식
* HTTP URL = ‘/api/boards/{board\_id}/{page}
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| board \_id | int | **O** | 게시글 카테고리 id |  |
| page | int |  | 페이지 네비게이션을 위한 파라미터 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| article\_id | 2 | 게시판 카테고리 id | Int |  |
| member\_id | 2 | 작성자 id | int | 작성자 구분용 (닉네임 중복 가능) |
| nickname | 2 | 작성자 닉네임 | String |  |
| title | 2 | 글 제목 | String |  |
| view | 2 | 조회수 | int |  |
| create\_date | 2 | 생성일 | int |  |
| thumbnail | 2 | 대표 이미지 사진 | String | URI(String) |

**2.3.3 게시글 상세 조회**

게시글 id를 통해 해당 게시글을 상세 조회한다.

* **Request**
* Method = GET방식
* HTTP URL = ‘/api/boards/detail/{article\_id}’
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| board\_id | int | **O** | 게시글 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| title | 1 | 글 제목 | String |  |
| content | 1 | 글 내용 | String |  |
| board\_id | 1 | 게시판 id | int |  |
| thumbnail | 1 | 숨김 여부 | boolean |  |
| view | 1 | 조회수 | int |  |
| member\_id | 1 | 유저 id |  | 작성자 구분용 (닉네임은 중복 가능) |
| nickname | 1 | 유저 닉네임 | String |  |
| create\_date | 1 | 글 작성 날짜 | String | Date 형태로 주어짐 |
| modify\_date | 1 | 글 수정 날짜 | String | Date 형태로 주어짐 |

**2.3.4 게시글 생성**

글 정보를 받아 글을 저장한다. token 인증을 통해 사용자를 확인한다.

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/boards’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| title | String | **O** | 게시글 제목 |  |
| content | String | **O** | 게시글 내용 |  |
| thumbnail | String | **O** | 대표 이미지 | 썸네일 URI |
| board\_id | int | **O** | 게시글 카테고리 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.5 게시글 수정**

글 정보를 받아 글을 수정한다. token 인증을 통해 사용자를 확인한다.

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/boards/{article\_id}’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| article\_id | int | **O** | 수정할 게시글 id |  |
| title | String | **O** | 게시글 제목 |  |
| content | String | **O** | 게시글 내용 |  |
| thumbnail | String | **O** | 대표 이미지 | 썸네일 URI |
| board\_id | int | **O** | 게시글 카테고리 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.6 게시글 삭제**

유저 정보화 함께 글 삭제를 요청한다. 해당 게시글을 작성한 사용자의 요청이에만 해당 게시글을 삭제 처리한다. 실제 삭제가 아닌 삭제 처리 요청(is\_delete)으로 데이터는 남아있다..

* **Request**
* Method = PUT방식
* HTTP URL = ‘/api/boards{article\_id}’
* Header Content-Type = application/json
* Parameter 형식(Header Content-Type) = application/json

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | 사용자 토큰 정보 | header로 전송 |
| article\_id | int | **O** | 게시글 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 |

**2.3.7 게시글 검색**

검색 기준과 검색어를 사용해 게시글을 검색한다..

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/boards/search
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| boards \_id | int | **O** | 게시글 카테고리 id |  |
| type | String | **O** | 검색 타입(title, view) | title/content/author 중 1가지로 검색 |
| content | String | **O** | 검색내용(검색어) |  |
| page | int | **O** | 페이지 네비게이션 번호 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| article\_id | 2 | 게시판 카테고리 id | Int |  |
| member\_id | 2 | 작성자 id | int | 작성자 구분용 (닉네임 중복 가능) |
| nickname | 2 | 작성자 닉네임 | String |  |
| title | 2 | 글 제목 | String |  |
| view | 2 | 조회수 | int |  |
| create\_date | 2 | 생성일 | int |  |
| thumbnail | 2 | 대표 이미지 사진 | String | URI(String) |

**2.3.8 이미지 저장**

이미지를 저장한다. 이미지를 저장한 후 해당 이미지 URI를 반환한다.

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/boards/image’
* Header Content-Type = multipart/form-data
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| article\_id | String | **O** | 게시글 제목 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.9 이미지 삭제**

해당 이미지를 삭제한다. 이미지 id를 통해 서버 및 db에서 삭제한다.

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/boards/image’
* Header Content-Type = multipart/form-data
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| article\_file\_id | String | **O** | 이미지 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.10 게시글 댓글 조회**

해당 게시글에 작성된 댓글 목록을 가져온다...

* **Request**
* Method = GET방식
* HTTP URL = ‘/api/boards/comment
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| article\_id | int | **O** | 게시글 id |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| comment\_id | 1 | 댓글 id | int |  |
| nickname | 1 | 작성자 닉네임 | String |  |
| create\_date | 1 | 댓글 생성일 | String |  |
| modify\_date | 1 | 댓글 수정일 | String |  |
| sub\_comments | 2 | 대댓글 목록 | List | 대댓글 목록 형태 sub\_comments {  comment\_id,  nickname  create\_date  modify\_date } |

**2.3.11 댓글 작성**

댓글 내용을 기입하여 새로운 댓글을 저장한다. token을 통해 댓글 작성자 정보를 조회한다.

* **Request**
* Method = POST방식
* HTTP URL = ‘/api/comments’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | token 정보 | header로 전달 |
| article\_id | int | **O** | 게시글 id |  |
| content | String | **O** | 댓글 내용 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.12 댓글 수정**

사용자의 댓글 내용을 수정한다. token 정보를 통해 수정 권한을 확인한다.

* **Request**
* Method = PUT 방식
* HTTP URL = ‘/api/comments’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | token 정보 | header로 전달 |
| article\_id | int | **O** | 게시글 id |  |
| content | String | **O** | 댓글 내용 |  |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |

**2.3.13 댓글 삭제**

사용자가 작성한 댓글을 삭제한다. token을 통해 삭제 권한을 확인한다.

* **Request**
* Method = DELETE 방식
* HTTP URL = ‘/api/comments’
* Header Content-Type = application/json
* Parameter 형식

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파라미터 명** | **타입** | **필수여부** | **설명** | **비고** |
| token | String | **O** | token 정보 | header로 전달 |

* **Response**
* 반환 타입 = **JSON 형태**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **depth** | **설명** | **값구분** | **비고** |
| status | 1 | 결과 확인값 | int | 1 : 성공 3 : 데이터 규칙 위반 4 : 데이터 부재 |