API 명세서

작성자 : 서울 2반 2팀

**목차 (Table of Contents)**

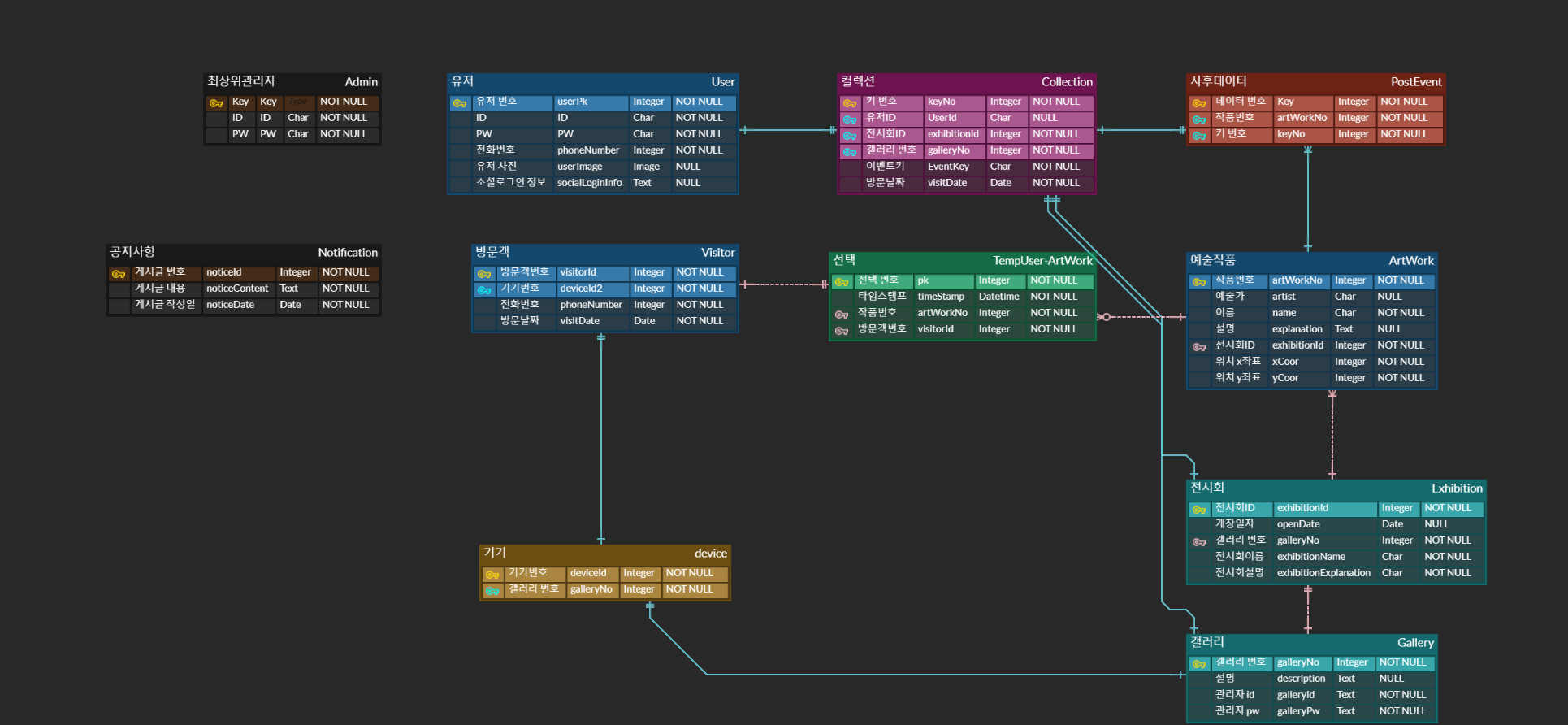
|  |  |
| --- | --- |
| 1. | 개요 및 프로젝트 목적 |
| 2. | 프로젝트 ERD 설계 |
| 3. | 명세 : 사용자 회원 관리 |
| 4. | 명세 : 갤러리 회원 관리(관리자) |
| 5. | 명세 : 사용자 – 전시부 |
| 6. | 명세 : 유저 – 선택 데이터 |
| 7. | 명세 : 유저 – 사후 데이터 |
| 8. | 명세 : 관리자 – 전시부 |

1. **개요 및 프로젝트 목적**

미술관 사용자들에게 관람시 원하는 미술품을 관람 중 편하게 리스트에 추가하고 리스트를 이용한 기념품을 제작(웹 페이지 형태로) 후 제공

1. **프로젝트 ERD**

ERD의 경우 gitlab이나 Notion에 공유된 이미지를 참조.



**3. 명세 : 사용자 회원 관리**

3-1) 회원가입

Request Method : POST

URI : /auth/signup

Authentication : None

Body : { “username” : “string”,   
“password” : “string”,   
“email” : “string”,   
“phoneNumber” : “bigint” }

Response :

HTTP 201 Created

{ “message” : “회원가입 완료”}

3-2) 로그인

Request Method : POST

URI : /auth/login

Authentication : None

Body : { “username” : “string”,   
“password” : “string”}

Response :

HTTP 200 OK

{ “accessToken” : “string”  
“refreshToken” : “string”}

3-3) 토큰 갱신

Request Method : POST

URI : /auth/refresh

Authentication : RefreshToken

Body : None

Response :

HTTP 200 OK

{ “accessToken” : “string” }

3-4) 로그아웃

Request Method : POST

URI : /auth/logout

Authentication : access token

Body : None

Response :

HTTP 200 OK

{ “message” : “로그아웃 성공”}

3-5) 사용자 정보 조회

Request Method : GET

URI : /users/me

Authentication : Access token

Response :

HTTP 200 OK

3-6) 사용자 정보 업데이트

Request Method : PUT

URI : /users/me

Authentication : Access token

Response : { “username” :

“phoneNumber” : }

HTTP 200 OK

3-7) 비밀번호 변경

Request Method : POST

URI : /users/me/change-password

Authentication : Access token

Body : { “oldPassword” :

“newPassword” : }

Response :

HTTP 200 OK

**4. 명세 : 갤러리 회원 관리(관리자)**

4-1) 갤러리 생성

Request Method : POST

URI : /admin/galleries

Authentication : Access token(Admin)

Body {

“galleryname” : “string”,

“description” : “string”

}

Response : {

“gallery” : {

“galleryname” : “string”,

“password” : “string” }

}

HTTP 200 OK

4-2) 갤러리 정보조회

Request Method : GET

URI : /galleries/{galleriesId}

Authentication : Access token(Admin or Gallery)

Response : {

“gallery” : {

“galleryname” : “string”,

“description” : “string” }

}

HTTP 200 OK

4-3) 갤러리 정보수정

Request Method : PUT

URI : /galleries/{galleriesId}

Authentication : Access token(Admin or Gallery)

Body : { “galleryname” : “string”,

“description” : “string”}

Response : {

“gallery” : {

“galleryname” : “string”,

“description” : “string” }

}

HTTP 200 OK

**5. 명세 : 사용자 – 기기**

5-1) 방문객 등록

Request Method: POST

URI: /devices/{deviceID}

Input Example: {“phoneNumber”: 8201012345678}

Output Example: HTTP 200 OK

설명: 사용자가 들고 있는 기기번호와 전화번호를 입력하면 class Visitor에 저장되는 함수이다.

5-2) 방문객 삭제

Request Method: DELETE

URI: /devices/{deviceID}

Input Example: 없음.

Output Example: HTTP 200 OK.

설명: 삭제 요청이 들어오면 기기에 등록되어 있는 전화번호를 바탕으로 기존 유저 데이터에 전화번호가 일치하는 유저가 있는지 파악한다. 그 다음 유저에게 이벤트 키를 발행하는 함수를 호출하여 이벤트 키를 생성한 후 이벤트 키를 문자로 발송하고 사후데이터에 키와 방문객이 고른 작품 데이터를 전송한다. 만일 기존 유저 데이터에 방문객이 존재한다면, 방문객의 개인 페이지에 이를 볼 수 있도록 한다.

**6. 명세 : 사용자 – 선택 데이터**

6-1) 방문객 작품 고르기

Request Method: POST

URI: /selections/devices/{deviceID}

Input Example: {“drawingID”:2}

Output Example: HTTP 200 OK

설명: 방문객이 들고 있는 기기의 추가 버튼을 누르면 현재 기기(방문객의 위치)의 위치가 속한 영역을 기준으로 해당 작품을 DB에 방문객이 고른 작품 목록에 추가해준다.

6-2) 방문객 작품 삭제

Request Method: DELETE

URI: /selections/devices/{deviceID}

Input Example: 없음.

Output Example: HTTP 200 OK

설명: 삭제 요청이 들어오면 가장 최근에 DB에 추가된 방문객이 고른 작품을 삭제해준다. 만일 리스트가 현재 빈 상태라면 삭제 요청을 무시한다

**7. 명세 : 사용자 – 사후 데이터**

7-1) 로그인 한 사용자의 키 값 반환

Request Method: GET

URI: /users/me/previousExhibitions

권한: 로그인 필요

Input Example: 없음

Output Example: {“ExhibitionList”: [{“ExhibitionKey”: “9u8tuh98u384u8”, “Exhibition”: “AA Exhibition”, “Gallery”: “Louvre Museum”, “visitDate”: “2019-03-01 12:34:00”}]}

설명: 로그인 한 유저는 개인 페이지에서 자신이 관람했던 전시회의 목록을 제공하며 전시회와 전시관의 이름 그리고 방문일시를 반환해준다.

7-2) 사후 데이터 조회

Request Method: GET

URI: /postDatas/{Key}

Input Example: 없음.

Output Example: {“ExhibitionID”: 1, “ExhibitionName”: “AA Exhibition”, “GalleryID”: 1, “GalleryName”: “Louvre Museum”, “visitDate”: “2019-03-01 12:34:00”, “WorkList”: “[{“paintName” : “Mona-Liza”, “paintPath”: “{paintPath}”}, {“paintName”: “Starry Night”, “paintPath”: “{paintPath2}”}]”}

설명: 사후 데이터 조회시 해당 키의 전시회, 전시관과 방문시각, 시간 순서대로 사용자가 고른 작품이 WorkList 배열 안에 Object 형태로 감싼 형태로 출력된다.

7-3) 사후 데이터 수정 요청 전 근처 작품 조회.

Request Method: GET

URI: /drawings/{drawingId}/nearby

Input Example: 없음

Output Example: {“NearbyDrawing”: [{“DrawingID”: 1, “DrawingName”: “Mona-Liza”, “DrawingUrl”: “”}]}

설명: 해당 작품 근처 작품을 요청할 경우 근처 작품을 최대 5개 포함한 리스트를 보내준다. 작품 개수가 5개 미만인 경우, 그만큼만 보내준다. (위의 Output Example에는 1개만 표시하였으나 실제로는 5개를 보내줄 것이다.)

7-4) 사후 데이터 수정 요청

Request Method: PUT

URI: /postData/{Key}

Input Example: {“OriginalDrawingID”: 3, “AfterDrawingID”: 5}

Output Example: {“msg” : “OK”}

설명: 3번 작품을 5번 작품으로 변경하고 싶은 경우 위의 Input Example처럼 요청을 한다.

7-5) 사후 데이터 삭제

Request Method: DELETE

URI: /postData/{Key}

Input Example: {“DrawingID”: 3}

Output Example: {“msg” : “OK”}

설명: 3번 작품을 목록에서 삭제하고 싶은 경우 위의 Input Example처럼 요청한다.

**8. 명세: 관리자 – 전시부**

8-1) 관리자 전시회 조회

Request Method: GET

URI: /exhibitions

Authorization: 관리자 로그인 필요

Input Example: 없음

Output Example: {“ExhibitionList”:[{“ExhibitionID”: 2, “ExhibitionName”: “BB Exhibition”, “OpenDate”: “2020-01-01 12:34:00”, “ExhibitionDescription”: “New Art Exhibition”}]}

설명: 관리자 갤러리의 전시회 리스트를 리턴해준다.

8-2) 관리자 전시회 추가.

Request Method: POST

URI: /exhibitions

Input Example: {“ExhibitionName”: “BB Exhibition”, “OpenDate”: “2020-01-01 12:34:00”, “ExhibitionDescription”: “New Art Exhibition”}

Output Example: {“msg”:”OK”}

설명: Input Example대로 전시관 이름, 개장 일자, 전시회 설명을 입력해주면 새로운 전시회를 등록해준다.

8-3) 관리자 그림 조회.

Request Method: GET

URI: /exhibitions/{exhibitionId}/drawings

Input Example: 없음.

Output Example: {“DrawingList”: [{“Name”: “New MonaLiza”, Artist”: “qwerty”, “LocationX”: 161, “LocationY”: 231, “Description”: “This is the hommage of the famous painting MonaLiza”}]}

설명: 전시회 번호를 입력하면 해당 전시회의 모든 그림의 목록을 반환한다.

8-4) 관리자 그림 추가.

Request Method: POST

URI: /exhibitions/{exhibitionId}/drawings

Input Example: {“DrawingName”: “New MonaLiza”, “Artist”: “qwerty”, “LocationX”: 161, “LocationY”: 231, “Description”: “This is the hommage of the famous painting MonaLiza”}

Output Example: {“msg” : “OK”}

설명: 관리자가 Input Example과 같은 쿼리를 보내면 Drawing DB에 해당 그림을 추가한다.