EnjoyTrip 설계서

프로젝트명 : EnjoyTrip

|  |  |
| --- | --- |
| **팀명** | REST |
| **팀장** | 김상민 |
| **팀원** | 김석주 |

목차

1. **테이블 구조도(ERD) 1**

**1-1. ERD**

**1-2. 각 테이블 명세**

1. [**클래스 다이어그램**](#_Toc84966411) **8**

**2-1. 기능별 클래스 다이어그램**

1. **화면 설계서 16**

**3-1. Vue 설계**

**제 1 장 테이블 구조도(ERD)**

**제 1 절. ERD**

텍스트, 스크린샷, 평행, 폰트이(가) 표시된 사진

자동 생성된 설명

**제 2 절. 각 테이블의 명세**

* alarm

어떤 사용자의 게시글에 댓글이 달리거나 혹은 다른 사용자에게 쪽지를 보냈을 때 게시글의 댓글은 해당 게시글을 작성한 사용자에게 또, 쪽지의 경우 수신자에게 알람 데이터를 저장해 놓는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| num | int autoIncrement | Surrogate key |
| sep | int | 해당 row의 데이터가 댓글에 대한 것인지 혹은 쪽지에 대한 것인지를 나타내기 위한 컬럼 |
| articleid | varchar(100) | 댓글인 경우 댓글이 달린 게시글 번호, 쪽지인 경우 해당 쪽지를 받는 수신인 |
| createtime | datetime | 해당 알람이 생성된 시간 |
| userid | varchar(100) | 댓글인 경우 해당 게시글을 쓴 사람, 쪽지인 경우 해당 쪽지를 받는 송신인 |

* attraction\_description

해당 관광지에 대한 정보를 저장한 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| content\_id | int | 해당 관광지를 나타내는 primary key |
| homepage | varchar(100) | 해당 관광지의 홈페이지 주소 |
| overview | varchar(10000) | 해당 관광지에 대한 간략한 설명 |
| telname | varchar(45) | 해당 관광지 연락처 |

* attraction\_detail

해당 관광지에 대한 정보를 저장한 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| content\_id | int | 해당 관광지를 나타내는 primary key |
| cat1 | varchar(3) |  |
| cat2 | varchar(5) |  |
| cat3 | varchar(9) |  |
| created\_time | varchar(14) | 해당 데이터 생성 시간 |
| modified\_time | varchar(14) | 해당 데이터 수정 시간 |
| booktour | varchar(5) |  |

* attraction\_info

해당 관광지에 대한 위치, 이미지, 연락처 등 다양한 정보를 저장한 테이블

|  |  |  |
| --- | --- | --- |
| attribute | Type | 의미 |
| content\_id | int | 해당 관광지를 나타내는 primary key |
| content\_type\_id | int | 해당 관광지가 포함되는 유형(쇼핑, 음식점 등등) |
| title | varchar(100) | 해당 관광지 이름 |
| addr1 | varchar(100) | 해당 관광지 주소 |
| addr2 | varchar(50) | 해당 관광지 주소(간략한 주소) |
| zipcode | varchar(50) | 지역코드 |
| tel | varchar(50) | 해당 관광지 연락처 |
| first\_image | varchar(200) | 해당 관광지 이미지 링크 |
| first\_image2 | varchar(200) | 해당 관광지 이미지 링크 |
| readcount | int | 해당 관광지 조회 수 |
| sido\_code | int | 해당 관광지가 포함된 시/도 코드 |
| gugun\_code | int | 해당 관광지가 포함된 구/군 코드 |
| latitude | decimal(20,17) | 해당 관광지 위도 |
| longitude | decimal(20,17) | 해당 관광지 경도 |
| mlevel | varchar(2) |  |
| win | int |  |

* board

4개의 게시판에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| num | int autoincrement | Surrogate key |
| createtime | datetime | 해당 글 작성 시간 |
| title | varchar(200) | 해당 글 제목 |
| content | varchar(2000) | 해당 글 내용 |
| writerid | varchar(100) | 해당 글 작성자 아이디 |
| hits | int | 해당 글 조회 수 |
| boardid | int | 어떤 게시판인지 알려주는 데이터  1 : 공지사항  2 : 여행후기게시판  3 : 자유게시판  4 : 일행구하기 |

* comment

댓글에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| num | int autoincrement | Surrogate key |
| articleno | int | 해당 댓글이 작성된 게시글 번호 |
| writerid | varchar(100) | 해당 댓글 작성자 |
| content | varchar(2000) | 해당 댓글 내용 |
| createtime | datetime | 해당 댓글 생성 시간 |

* electric\_charger

전기차 충전소에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| no | int autoincrement | Surrogate key |
| statId | char(8) | 해당 전기차 충전소 ID |
| chgerId | char(2) | 해당 전기차 충전소 충전기 ID |
| chgerType | char(2) | 해당 전기차 충전소 충전기 타입 |

* electric\_charing\_station

전기차 충전소 자체에 대한 세부 정보를 저장하고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| no | int autoincrement | Surrogate key |
| statId | char(8) | 전기차 충전소 아이디 |
| statNm | varchar(100) | 전기차 충전소 이름 |
| addr | varchar(150) | 전기차 충전소 주소 |
| location | varchar(200) |  |
| lat | decimal(15,12) | 해당 전기차 충전소가 위치한 위도 |
| lng | decimal(15,12) | 해당 전기차 충전소가 위치한 경도 |
| coordinate | point | 두 좌표 사이의 거리를 구하기 위해 만든 데이터 |

* gugun

구/군에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| gugun\_code | int | 구/군 코드 |
| gugun\_name | varchar(30) | 구/군 이름 |
| sido\_code | int | 해당 구/군이 포함된시/도 코드 |

* note

쪽지에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| num | int autoincrement | Surrogate key |
| title | varchar(200) | 쪽지 제목 |
| receiver | varchar(50) | 수신자 ID |
| sender | varchar(50) | 송신자 ID |
| content | varchar(300) | 쪽지 내용 |
| createtime | datetime | 해당 쪽지가 생성된 시간 |
| open | tinyint(1) | 해당 쪽지를 수신자가 열었는지 아닌지 판단하는 데이터 |

* path

사용자가 작성한 여행 경로 계획을 저장하는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| id | int autoincrement | Surrogate key |
| user\_id | varchar(100) | 해당 경로 계획을 짠 사용자 Id |
| path\_id | int | 동일한 사용자 내에서 경로 계획을 구분하기 위한 데이터 |
| title | varchar(200) | 해당 여행 계획 제목 |
| sequence | int | 해당 여행 경로에서 해당 관광지를 방문하는 순서 |
| content\_id | int | 해당 여행 경로에서 해당 순번에 방문하려고 계획한 관광지 |

* sido

시/도에 대한 정보를 저장하고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| sido\_code | int | 시/도 코드 |
| sido\_name | varchar(30) | 시/도 명 |

* user

사용자에 대한 정보를 담고 있는 테이블

|  |  |  |
| --- | --- | --- |
| Attribute | Type | 의미 |
| user\_id | varchar(100) | 사용자 아이디 |
| name | varchar(100) | 사용자 이름 |
| pass | varchar(100) | 사용자 비밀번호 |
| email | varchar(100) | 사용자 이메일 |
| path\_cnt | int | 사용자가 지금까지 생성한 여행 계획 수 |
| token | varchar(1000) | Refresh token 데이터 |

**제 2 장 클래스 다이어그램**

**제 1절. 기능별 클래스 다이어그램(Controller 메소드에 대한 설명만을 작성했습니다.)**

**● Board**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| List<Board> list(String) | 사용자 아이디를 입력 받아서 해당 사용자가 작성한 게시글의 정보만을 받아오는 메소드 |
| Board Modify(Board) | 수정된 게시글의 정보를 받아서 db에 수정된 게시글의 정보를 저장하는 메소드 |
| Board detail(int) | 게시글의 키값을 받아서 해당 게시글 정보를 반환하는 메소드 |
| Board insert(Board) | 등록할 게시글의 정보를 받아서 db에 저장하는 메소드 |
| List<Board> Board list(int) | 어떤 게시판인지 boardtype에 대한 정보를 매개변수로 받아서 해당 게시판에 등록된 글의 정보를 반환하는 메소드 |
| List<Board> list(SearchInfo) | 제목,내용,제목+내용 중 어떤 것으로 어떤 검색어를 가지고 검색을 할지에 대한 정보를 받아서 그에 해당하는 게시글에 대한 정보를 반환하는 메소드 |
| int delete(int) | 게시글의 num 값을 받아서 해당하는 게시글을 삭제하는 메소드 |

**● Charger**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| ResponseEntity<Map<String, Object>> getCharger(ChargerDto) | 관광지의 위치 정보를 받아서 그에 대한 위치 정보를 통해서 인근 2km 이내에 있는 전기차 충전소 리스트를 반환하는 메소드 |

**● Comment**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| List<Comment> list(int) | 글 번호를 받아서 해당 글번호에 달린 댓글 리스트를 반환하는 메소드 |
| int delete(int) | 댓글 번호를 받아서 댓글을 삭제하는 메소드 |
| Comment insert(Comment) | 댓글 정보를 받아서 해당 댓글 정보를 db에 저장하는 메소드 |
| Comment update(Comment) | 수정된 댓글 정보를 받아서 수정된 댓글 정보를 db에 저장하는 메소드 |

**● WebMvcConfiguration, ConfirmInterceptor**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| void addInterceptor(InterceptorRegistry) | 로그인 하기 전에 토큰 검사를 하는 interceptor를 등록하는 메소드 |
| boolean preHandle(HttpServletRequest, HttpServletResponse, Object) | 로그인 하기 전에 토큰 검사를 하는 메소드 |

**● Alarm**

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| int insert(Alarm) | 알람 정보를 받아서 db에 등록하는 메소드 |
| int deleteComment(String) | 게시글 아이디를 받아서 해당 게시글에 등록된 알람을 제거하는 메소드 |
| int getCount(String) | 사용자 아이디를 입력받아서 해당 사용자아이디로 등록된 알람 수를 세는 메소드 |
| List<Alarm> list(String) | 사용자 아이디를 입력받아서 해당 사용자 아이디로 등록된 알람 정보 리스트를 가져오는 메소드 |
| int deleteNote(String) | 사용자 아이디를 받아서 사용자 아이디로등록된 쪽지 알람을 제거하는 메소드 |

**● Note**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| List<Note> getSendList(String) | 사용자 아이디를 입력 받아서 해당 사용자가 보낸 쪽지 리스트를 반환하는 메소드 |
| int modify(Note) | 사용자가 수정한 쪽지 내용을 받아서 이를db에 저장하는 메소드 |
| List<Note> getReceiverList(String) | 사용자 아이디를 받아서 사용자가 받은 쪽지 리스트를 반환하는 메소드 |
| int getNoteCount(String) | 사용자 아이디를 입력받아서 사용자가 아직 확인하지 못한 쪽지 개수를 반환하는 메소드 |
| Note insert(Note) | 쪽지 정보를 입력받아서 이를 db에 저장하는 메소드 |
| Note getNote(int) | 쪽지 번호를 입력받아서 해당 쪽지 번호에해당하는 쪽지 정보를 반환하는 메소드 |

**● Path**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| ResponseEntity<?> insertPath(List<Path>, HttpServletRequest) | 사용자가 등록할 여행 계획 정보를 입력받아서 이를 db에 저장하는 메소드 |
| ResponseEntity<?> listPath(User, HttpServletRequest) | 사용자 정보를 받아서 해당 사용자 정보로 등록된 모든 여행 계획 정보를 반환하는 메소드 |
| int delete(String, int) | 사용자 아이디와 사용자가 삭제할 여행 경로 아이디를 입력받아서 해당 여행 경로를 삭제하는 메소드 |

**● Place**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| ArrayList<Place> getPlace(int, int, int, String) | 시/도 코드, 구/군 코드, 관광지 유형 코드, 검색어를 입력받아서 해당 관광지 리스트를 반환하는 메소드 |
| ArrayList<GugunDto> getGugun(int) | 시/도 코드를 입력받아서 해당 시/도에 포함된 구/군 리스트를 반환하는 메소드 |
| ArrayList<SidoDto> getSido() | db에 저장된 시/도 정보를 반환하는 메소드 |

**● User**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 메소드 | 기능 |
| ResponseEntity<Map<String, Object>> getInfo(String, HttpServletRequest) | 사용자 아이디를 입력 받아서 토큰의 유효성 검사 후 사용자의 정보를 반환하는 메소드 |
| ResponseEntity<?> join(User, HttpServletRequest) | 회원가입 할 사용자의 정보를 입력받아 db에 저장하는 메소드 |
| ResponseEntity<Map<String, Object>> changePass(User) | 바뀐 사용자의 비밀번호를 받아서 수정하는 메소드 |
| ResponseEntity<Map<String, Object>> modify(User) | 수정될 사용자 정보를 받아서 수정될 사용자 정보로 db 데이터를 바꾸는 메소드 |
| ResponseEntity<?> removeToken(String) | 사용자 아이디를 받아서 해당 사용자의 refresh token 정보를 삭제하는 메소드 |
| User checkId(String) | 아이디로 사용하고 싶은 문자열을 입력받아 중복 검사를 하는 메소드 |
| ResponseEntity<?> passCheck(User) | 사용자 정보를 받아서 해당 사용자 정보로등록된 사용자가 있는지 확인하는 메소드 |
| ResponseEntity<?> refreshToken(User, HttpServletRequest) | Access token이 만료되어 refresh token을 검사하는데 refresh token이 아직 만료되지 않았다면 정상적으로 access token을 발급해주고 refresh token이 만료되었다면 UnAuthorized 예외를 던지는 메소드 |
| ResponseEntity<Map<String, Object>> login(User) | 사용자 정보를 받아서 해당 사용자가 있는지 확인하고 있다면 access, refresh 토큰을 발급해서 넘겨주는 메소드 |

**제 3 장 화면 설계서**

**제 1절 기능별 Vue 설계**

**● App.Vue**

텍스트, 물, 스크린샷, 하늘이(가) 표시된 사진

자동 생성된 설명

App.vue는 위에서부터 순차적으로 TheHeader, TheNav, (TheHeader, TheNav)에서 선택한 것에 따라 보이는 router-view(현재는 HomeView), TheFooter로 구성된다.

**● The Header**

- 로그인 전



- 로그인 후

텍스트, 폰트, 라인, 화이트이(가) 표시된 사진

자동 생성된 설명

로그인 전에는 회원가입과 로그인이 보이다가 로그인 후에는 로그인 계정에 따른 메시지와 사람 아이콘과 알림 아이콘이 보이게 된다.

**● TheHeader, TheNav에서 클릭한 것에 따른 router view의 변화**

- TheHeader의 요소를 클릭한 경우

- 회원가입(UserRegister.vue)

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

- 로그인(UserLogin.vue)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

회원가입과 로그인 선택시 각각의 정보를 입력하도록 하는 화면으로 router-view 바뀌게 된다.

- 로그인 후 내 정보 보기를 선택한 경우(MyPage.vue)

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

왼쪽의 MyPageList.vue에서 선택한 것에 따라서 오른쪽의 화면이 바뀌게 된다.

회원정보를 선택한 경우 위와 같은 화면이 나오게 되고 게시글 관리, 내 여행 루

트, 쪽지를 선택하면 아래와 같은 화면이 나오게 된다.

- 로그인 후 알람을 선택한 경우(AlarmList.vue)

텍스트, 폰트, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

알람정보를 AlarmList.vue에서 출력해주고 하나하나의 요소는 AlarmItem.vue에서 관리해준다. 쪽지에 대한 알람을 클릭할 경우 쪽지 페이지로 이동하게 되고 댓글 알람을 클릭할 경우 해당 게시글로 이동하게 되고 해당 알람은 사라지게 된다.

* 게시글 관리 선택

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명게시글 관리에서는 MyBoardContents.vue를 통해서 작성한 게시글 목록을 보여주고 글 하나하나는 BoardListItem.vue를 통해서 table로 보여준다.

* 내 여행 루트 선택

텍스트, 나무, 스크린샷이(가) 표시된 사진

자동 생성된 설명MyTripRoute.vue에서 자신이 이전에 계획했던 여행 계획을 MyTripRouteItem.vue를 통해서 카드형식으로 하나하나 확인하고 정보보기를 통해서 수정, 삭제 버튼을 통해서 삭제할 수 있다.

* 쪽지 선택(MyNote.vue)

텍스트, 라인, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명MyNote.vue에서 보낸쪽지함을 선택하는 경우 SendNote.vue가 보여지고 받은쪽지함을 선택하는 경우 ReceiveNote.vue가 보여지게 된다. 그리고 각각의 요소는 NoteSendItem.vue와 NoteReceiveItem.vue로 보여지게 된다.

- TheNav의 요소를 클릭한 경우

- 테마여행(TheSearchPlace.vue)

텍스트, 지도, 스크린샷, 아틀라스이(가) 표시된 사진

자동 생성된 설명

TheSearchPlace.vue는 시/도를 선택하는 SelectSido.vue, 구/군을 선택하는 SelectGugun.vue, kakaomap을 보여주는 KakaoMap.vue, 검색한 장소들을 카드로 보여주는 PlaceList.vue로 구성된다. 각각의 카드 하나씩은 PlaceListItem.vue로 구성된다.

- 나의 여행 계획(TheMakePath.vue)

텍스트, 지도, 스크린샷, 아틀라스이(가) 표시된 사진

자동 생성된 설명

TheMakePath.vue는 시/도를 선택하는 SelectSido.vue, 구/군을 선택하는 SelectGugun.vue, kakaomap을 보여주는 PathKakao.vue, 검색하고 마커를 통해 여행경로에 추가한 장소들의 정보를 보여주는 PathList.vue로 구성된다.

- 게시판(TheBoard.vue)

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

TheBoard.vue는 좌측에 어떤 게시판을 볼 지 선택할 수 있는 BoardType.vue, 선택한 BoardType에 맞는 내용을 화면에 보여주는 NoticeBoard.vue, ReviewBoard.vue, FreeBoard.vue, TeamBoard.vue로 구성된다. 그리고 각 게시판의 페이징을 처리하는 컴포넌트인 BoardPage.vue가 각 게시판 vue에 포함된다.

* 댓글(CommentList.vue)

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

게시글을 클릭해서 들어가보면 제일 위에 BoardView.vue 컴포넌트를 통해서 게시글 정보가 보이고 목록, 수정, 삭제 버튼을 통해서 각각의 기능을 수행할 수 있다. 이후 아래의 CommentWrite.vue를 통해서 댓글을 작성 후 등록할 수 있다. 마지막으로 아래의 CommentList.vue를 통해서 댓글 정보를 확인할 수 있고 댓글 하나하나는 CommentListItem.vue에서 관리하면 수정 및 삭제가 가능하다.