

2-3. (중급자를 위한) express를 이용하여 서버 구현 (route-controller-model)

⊙ 주차	2주차
⊙ 카테고리	과제
≡ 태그	
※ 상태	진행 중

주의

- model은 DB 연결이 아닌 JSON 더미 데이터로 진행
- 다음 과제를 위해 프로젝트 분리

인증, 인가와 관련된 기능은 구현하지 않았습니다.

- 인증된 사용자만 게시물 작성
- 본인이 작성한 게시물만 수정 가능 등

별도의 전달 객체를 만들지 않았습니다.

사용된 JSON의 구조는 아래와 같습니다.

```
{
  "sequence": 1,
  "users": [
    {
      "id": 0,
      "profileImagePath": "",
      "email": "",
      "password": "",
      "nickname": ""
    }
  ]
}
```

```
{
  "sequence": 1,
  "post": [
    {
      "id": 0,
      "author": "",
      "postImagePath": "",
      "title": "",
      "content": ""
    }
  ]
}
```

```
{
  "sequence": 1,
  "comments": [
    {
      "id": 0,
      "postId": "",
      "author": "",
      "content": ""
    }
  ]
}

{
  "message": "INVALID_EMAIL"
}
```

구현된 코드가 많아서 코드는 깃허브 [링크](#)로 대체했습니다.

API 명세서

- 유효성 검사에 사용된 정규식
 - email: /^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,}\$/
 - 알파벳, 숫자, @, . 만 가능
 - password: /^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[A-Za-z0-9]).{8,20}\$/
 - 8~20 글자, 대문자-소문자-숫자-특수문자 최소 1개씩 포함
 - nickname: /^[S]{1,10}\$/
 - 1~10 글자
 - image

- 한개의 파일만 허용
- title: /^{1,26}\$/
 - 1~26 글자
- content: /.+/
 ▪ 최소 1글자
- comment: /.+/
 ▪ 최소 1글자

URL	method	Content-Type	request body	response status code	response body	설명
로그인						
/api/sign-in	post	application/x-www-form-urlencoded	{ email: 텍스트, 필수 password: 텍스트, 필수 }	400	{ "message": "INVALID_EMAIL" }	유효하지 않은 이메일
				400	{ "message": "INVALID_PASSWORD" }	유효하지 않은 비밀번호
				200		로그인 성공
회원 가입						
/api/sign-up	post	multipart/form-data	{ profile-image: 파일, 필수 email: 텍스트, 필수 password: 텍스트, 필수 nickname: 필수 }	400	{ "message": "INVALID_PROFILE_IMAGE" }	유효하지 않은 프로필 사진
				400	{ "message": "INVALID_EMAIL" }	유효하지 않은 이메일
				400	{ "message": "INVALID_PASSWORD" }	유효하지 않은 비밀번호
				400	{ "message": "INVALID_NICKNAME" }	유효하지 않은 닉네임
				409	{ "message": "DUPLICATE_EMAIL" }	중복된 이메일
				409	{ "message": "DUPLICATE_NICKNAME" }	중복된 닉네임
				200		회원가입 성공
회원 정보 수정						
/api/users/profile-image-and-nickname	patch	multipart/form-data	{ profile-image: 파일, 필수 id: 텍스트, 필수 nickname: 텍스트, 필수 }	400	{ "message": "INVALID_PROFILE_IMAGE" }	유효하지 않은 프로필 사진
				400	{ "message": }	유효하지 않은 닉네임

URL	method	Content-Type	request body	response status code	response body	설명
					"INVALID_NICKNAME"	
				400	{ "message": "USER_NOT_FOUND" }	회원 없음
				409	{ "message": "DUPLICATE_NICKNAME" }	중복된 닉네임
				200		회원 정보 수정 성공
/api/users/password	patch	application/x-www-form-urlencoded	{ id: 텍스트, 필수 password: 텍스트, 필수 }	400	{ "message": "INVALID_PASSWORD" }	유효하지 않은 비밀번호
				400	{ "message": "USER_NOT_FOUND" }	회원 없음
				200		회원 정보 수정 성공
게시글 목록 조회						
/api/posts	get	none		200	[{ "id": 0, "author": "", "postImagePath": "", "title": "", "content": "" }, ...]	게시글 목록 조회 성공
게시글 상세 조회						
/api/posts/{id}	get	none		400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				200	{ "id": 0, "author": "", "postImagePath": "", "title": "", "content": "" }	게시글 상세 조회 성공
게시글 추가						
/api/posts	post	multipart/form-data	{ post-image: 파일, 선택 title: 텍스트, 필수 content: 텍스트, 필수 }	400	{ "message": "INVALID_TITLE" }	유효하지 않은 제목
				400	{ "message": "INVALID_CONTENT" }	유효하지 않은 내용
				200		게시글 추가 성공
게시글 수정						
/api/posts/{id}	patch	multipart/form-data	{ post-image: 파일, 선택 title: 텍스트, 필수 content: 텍스트, }	400	{ "message": "POST_NOT_FOUND" }	게시글 없음

URL	method	Content-Type	request body	response status code	response body	설명
			필수 }			
				400	{ "message": "INVALID_TITLE" }	유효하지 않은 제목
				400	{ "message": "INVALID_CONTENT" }	유효하지 않은 내용
				200		게시글 수정 성공
게시글 삭제						
/api/posts/{id}	delete	none		400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				200		게시글 삭제 성공
댓글 확인						
/api/posts/{postId}/comments	get	none		400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				200		게시글의 댓글 조회 성공
댓글 추가						
/api/posts/{postId}/comments	post	application/x-www-form-urlencoded	{ content: 텍스트, 필수 }	400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				400	{ "message": "INVALID_COMMENT" }	유효하지 않은 댓글
				200		댓글 추가 성공
댓글 수정						
/api/posts/{postId}/comments	patch	application/x-www-form-urlencoded	{ id: 텍스트, 필수 content: 텍스트, 필수 }	400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				400	{ "message": "COMMENT_NOT_FOUND" }	댓글 없음
				400	{ "message": "INVALID_COMMENT" }	유효하지 않은 댓글
				200		댓글 수정 성공
댓글 삭제						
/api/posts/{postId}/comments	delete	application/x-www-form-urlencoded	{ id: 텍스트, 필수 }	400	{ "message": "POST_NOT_FOUND" }	게시글 없음
				400	{ "message": "COMMENT_NOT_FOUND" }	댓글 없음
				200		댓글 삭제 성공