



응알응알. 맡겨주세요!

개발 방법론

TDD 기반 개발 방법론 채택

- 프로젝트의 요구 사항을 분석
- API 명세서 작성
- 테스트 코드 작성
- 실제 코드 작성
- 코드 평가



API 명세서 작성

10월 17일 편집

관

공유

구분	HTTP	Aa 기능	JWT	API Path	설명	개발 현황	테스트 성공
				t_id}			
게시판	POST	Create post heart	✓	/phheart/create	부모가 게시판에 하트를 누른다.	완료	✓
게시판	DELETE	Delete post heart	✓	/phheart/delete	부모가 게시판에 누른 하트를 취소함한다.	완료	✓
게시판	GET	Has script	✓	/pscript/hasScript/{post_id}	스크랩 했는지 확인	완료	□
게시판	POST	Create post script	✓	/pscript/create	부모가 게시물을 script한다.	완료	✓
게시판	DELETE	Delete post script	✓	/pscript/delete	부모가 게시판을 스크립트한 내역을 취소한다.	완료	✓
게시판	POST	Get post main content	✓	/post/main/create	게시판 메인 페이지 내용을 가져온다.	완료	✓
게시판	GET	Get poster profile	✓	/post/poster/profile/{parent_id}	포스터 작성자의 프로필 화면을 가져온다	완료	✓
게시판	POST	Recommend post	✓	/post/search/recommend	추천 페이지 생성한다.	완료	✓
게시판	POST	post search	✓	/post/search/result	제목 검색어로 게시물을 가져온다.	완료	✓
게시판	PUT	Update post photo	✓	/post/photoUpdate/{post_id}	post에 들어있는 photo를 수정한다.	완료	□
게시판	POST	Manage post script	✓	/pscript/	유저가 누른 스크립트가 데이터에 없으면 추가 있으면 삭제한다.	완료	□
게시판	POST	Manage post heart	✓	/phheart/	유저가 누른 하트가 데이터에 없으면 추가 있으면 삭제한다.	완료	□

+ 새 페이지

값 82



API 명세서 작성 - 세부내용

Create

구분

설명

API Path

HTTP

☒ JWT

개발 현황

☒ 테스트 성공

담당자

프론트 개발 현황

속성 3개 추가

Request

```
curl --location --request POST 'http://0.0.0.0' \
--header 'Content-Type: application/json' \
--data-raw '{
  "parent_id": "string",
  "password": "string",
  "email": "string",
  "nickname": "string",
  "signInMethod": "string",
  "emailVerified": 1,
}'
```

Response

```
{
  "parent": {
    "parent_id": "string",
    "password": "string",
    "email": "string",
    "nickname": "string",
    "signInMethod": "string",
    "emailVerified": 1,
  }
}
```

Exception 1:

email을 이미 존재하는 id 값으로 설정된 경우 에러를 반환.

Request

```
curl --location --request POST 'http://0.0.0.0:7701/parent' \
--header 'Content-Type: application/json' \
--data-raw '{
  "parent_id": "string",
  "password": "string",
  "email": "중복된 email 값",
  "nickname": "string",
  "signInMethod": "string",
  "emailVerified": 0
}'
```

Response

```
{
  "status_code": 406,
  "detail": "email already exists"
}
```



테스트 코드 작성

```
def test_create_parent(client, test_jwt):  
    response = client.post(  
        "/parent",  
        json=test_create_data  
    )
```

```
    assert response.status_code == 201  
    response_json = response.json()  
    assert "parent_id" in response_json
```

```
    assert  
    assert  
    assert  
    assert  
    assert  
    assert
```

```
    assert  
    assert
```

```
def test_create_parent_badtoken(client):  
    response = client.post(  
        "/parent",  
        json=test_create_data  
    )  
  
    assert response.status_code == 406  
    assert response.json() == {"detail": "parent_id already exists"}
```



실제 코드 작성: MVC 기반

```
class Parent(BaseModel):
    parent_id: int
    password: str
    email: str
    name: Optional[str]
    nickname: str
    gender: Optional[str]
    signInMethod: str
    emailVerified: bool
    photoId: Optional[int]
    description: Optional[str]
    mainAddr: str
    subAddr: Optional[str]
    hashList: List[str]

    class Config:
        orm_mode = True
        use_enum_values = True

    @router.post("/")
    def create_parent(createParentInput: CreateParentInput):
        """
        부모 생성
        --input
        - createParentInput.parent_id
        - createParentInput.email
        - createParentInput.password
        - createParentInput.nickname
        - createParentInput.signInMethod
        - createParentInput.emailVerified
        --output
        - parent: 부모 정보
        - x-jwt: JWT 토큰
        """
        try:
            parent = parentService.create_parent(createParentInput)

            if parent is None:
                raise HTTPException(status_code=HTTP_400_BAD_REQUEST, detail="Invalid input")

        except CustomException as e:
            raise HTTPException(status_code=e.status_code, detail=e.detail)
```

```
class ParentService:
    """
    부모 생성
    """
    def create_parent(self, createParentInput: CreateParentInput):
        """
        부모 생성
        --input
        - createParentInput.parent_id
        - createParentInput.email
        - createParentInput.password
        - createParentInput.nickname
        - createParentInput.signInMethod
        - createParentInput.emailVerified
        --output
        - parent: 부모 정보
        """
        db = get_db_session()

        # 부모 아이디 중복 확인
        error = db.query(Parent).filter(Parent.parent_id == createParentInput.parent_id).first()
        if error:
            raise HTTPException(status_code=HTTP_400_BAD_REQUEST, detail="Parent ID already exists")

        # 비밀번호 암호화
        if createParentInput.signInMethod == 'email':
            createParentInput.password = bcrypt.hashpw(
                createParentInput.password.encode('utf-8'), bcrypt.gensalt())

        # 이메일 중복 확인
```

```
class CreateParentInput(BaseModel):
    parent_id: int
    password: str
    email: str
    nickname: str
    signInMethod: str
    emailVerified: bool
```

```
class CreateParentOutput(BaseModel):
    parent: Optional[Parent]
    x_jwt: XJWT
```



돈든한 육아 도우미
응알응알. 맡겨주세요!

일정 관리 - 엑셀, Notion

캡스톤 일정표															
자동 저장															
삽입 그리기 페이지 레이아웃 수식 데이터 검토 보기 자동화 개발 도구															
맑은 고딕 (본문) 12 가 가															
가 가 간															
클립보드 글꼴 맞춤															
51 fx															
No.	개발 구분	내용 대분류	내용 소분류	내용	담당자	27주	28주	29주	30주	31주	32주	33주	34주	35주	
1															
2															
3		개발 환경 세팅	FastAPI 개발 환경 세팅	MVC 기반의 FastAPI 개발 환경 세팅	곽재원										
4			DB 세팅 및 명세서 작성	MySQL 테이블 생성 및 Prisma를 이용한 FastAPI와 연결	정가현										
5		유저(Parent) API	유저 CRUD API 및 Service 개발 및 테스트 코드 작성	유저가 입력한 이메일과 패스워드를 검사하여 로그인 처리를 하는 API 및 Service 개발	홍승조										
6		유저와 보안 API	유저 로그인(Login) API	사용자가 입력한 이메일과 패스워드를 검사하여 로그인 처리를 하는 API 및 Service 개발	홍승조										
7			사용자 관계 API	사용자 간의 친구 관계 생성 및 삭제하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
8		아기 API	아기(baby) API	아기 CRUD API 및 Service 개발 및 테스트 코드 작성	정가현										
9			게시물(Post) API	게시물 CRUD API 및 Service 개발 및 테스트 코드 작성	정가현										
10			좋아요(PHeart) API	게시물에 대한 좋아요 생성, 삭제 API 및 Service 개발 및 테스트 코드 작성	정가현										
11			스크립트(Pscript) API	게시물에 대한 스크립트 생성, 삭제 API 및 Service 개발 및 테스트 코드 작성	정가현										
12			조회(Pview) API	게시물에 대한 조회 생성, 삭제 API 및 Service 개발 및 테스트 코드 작성	정가현										
13			댓글(PComment) API	게시물에 대한 댓글과 대댓글 CRUD API 및 Service 개발 및 테스트 코드 작성	정가현										
14			댓글 좋아요(PCHeart) API	게시물 댓글에 대한 좋아요 생성, 삭제 API 및 Service 개발 및 테스트 코드 작성	정가현										
15			오늘의 게시물(Postmain) API	오늘의 게시물 및 사용자 맞춤 게시물 API 및 Service 개발 및 테스트 코드 작성	홍승조										
16			오버뷰(Overview) API	짜꿍, 친구들, 이야기 수 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
17			사용자 관계 관리 API	짜꿍들과 친구들의 정보를 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
18			조회한 게시물(MyViews) API	조회한 게시물들을 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
19			스크립한 게시물(MyScripts) API	스크립한 게시물들을 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
20			좋아한 게시물(MyLikes) API	좋아한 게시물들을 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
21			작성한 게시물(MyPost) API	유저가 작성한 게시물들을 반환하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
22		채팅 API	채팅(chatting) API	유저와 유저 간의 채팅 API 및 Service 개발	홍승조										
23			채팅방(chatroom) API	채팅방을 생성, 유저 초대 등 전반적인 채팅방 API 및 Service 개발 및 테스트 코드 작성	홍승조										
24		산모수첩 API	수첩 관리 API	산모 수첩의 생성, 수정, 삭제, 등을 관리하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
25			일기 관리 API	일기의 생성, 수정, 삭제, 등을 관리하는 API 및 Service 개발 및 테스트 코드 작성	정가현										
26			AI 의사 채팅 API	LLM와 API를 연결하여 AI 의사 답변을 반환하는 코드 작성	홍승조										
27		AI 의사 API	채팅방 관리 API	채팅방을 생성, 유저 초대 등 전반적인 채팅방 API 및 Service 개발 및 테스트 코드 작성	홍승조										
28			채팅 기록 API	채팅 기록을 관리, 삭제, 추출하는 API 및 Service 개발 및 테스트 코드 작성	홍승조										
29		울음 분석	울음 분석 API	울음 분석하는 AI의 결과를 반환하는 API 및 Service 개발 및 테스트 코드 작성	곽재원										
30			울음 패턴 분석 API	울음 패턴을 분석한 결과를 반환하는 API 및 Service 개발 및 테스트 코드 작성	곽재원										
31			울음 기록 API	울음 기록을 검색하고 반환하는 API 및 Service 개발 및 테스트 코드 작성	곽재원										
32		개발 환경 세팅	개발 환경 설정	MVC 구조를 가지도록 개발 환경과 폴더 구조를 설정함	곽재원										
33			인공지능 환경 설정	on-device AI를 위한 Java 환경 설정	곽재원										
34			firebase를 통한 보안	firebase를 통한 소셜 로그인 구현 및 보안 세션 활성화	곽재원										
35		보안 및 유저 인증	로그인, 회원가입 페이지	로그인 및 회원가입 페이지 생성	곽재원										
36			유저 상태관리	유저 정보 상태 관리 시스템 구성	곽재원										
37			설정 메인 페이지	설정 메인 페이지 레이아웃 제작	곽재원										
38			아기, 부모 정보 생성, 수정 페이지	부모 정보를 수정, 아기 정보를 생성, 수정하는 페이지 제작	곽재원										
39			이야기 관리 페이지	이야기 관리(보기, 선택 삭제, 모두 삭제) 페이지 구성	곽재원										
40			친구, 짜꿍 관리 페이지	부모 간의 관계(친구, 짜꿍) 관리 페이지 생성	곽재원										
41			오늘의 게시물 페이지	유저 추천 게시물, 등 게시물 리스트를 종합적으로 보여주는 페이지 생성	곽재원										
42			게시물 페이지	글과 이미지를 보여주는 게시물 페이지 생성	곽재원										
43			게시물 작성 페이지	글과 사진을 첨부하여 게시물을 작성하는 페이지 생성	곽재원										
44		어플리케이션	유저 프로필 페이지	유저의 게시물을 모아 보여 유저 간의 관계(친구)를 설정할 수 있는 프로필 페이지 생성	곽재원										

대상을 선택한 다음 <Enter> 키를 누르거나 [붙여넣기]를 선택합니다.

TODO

가현

- createBaby: babyId는 uuidv4로 생성하도록 변경.
 - baby 프로필사진 저장하는 폴더 관리.
 - test code 작성.
- setting 연결할 때 상담

승조

- postmain test code 확인.
- chat Docs 생성.
- SQL 생성 ← parent, baby, friend, post

재원

- friends API docs.



튼튼한 육아 도우미
응알응알. 맡겨주세요!

Git: Trunkbased 개발

