

샌드위치 API 만들기

1. 해당 Repository의 URL 기입

https://github.com/sunhye-kim/sandwich_store.git

2.API요청 및 응답처리과정기술

프로젝트 시작 시 호출할 명령어

```
python3 manage.py makemigrations inventory_management/  
python3 manage.py migrate inventory_management/  
python3 manage.py runserver
```

Swagger : <http://127.0.0.1:8000/swagger/>

(1) [POST] inventory/set_sandwich_ingredient_inventory/

용도	샌드위치 재료 데이터 INSERT API (type, name) 에 해당하는 데이터가 있는 경우 : 업데이트 - remain_cnt에 해당하는 값만큼 기존 remain_cnt 값에 더하기 - price 값 업데이트 (type, name) 에 해당하는 데이터가 없는 경우 : INSERT
테스트 url	http://127.0.0.1:8000/inventory/set_sandwich_ingredient_inventory/
테스트 curl	curl -X POST "http://127.0.0.1:8000/inventory/set_sandwich_ingredient_inventory/" -H "accept: application/json" -H "Content-Type: application/json" -H "X-CSRFToken: ?" -d "{ \"type\": \"토핑\", \"name\": \"치킨\", \"remain_cnt\": 100, \"price\": 1500}"
body	{ "type": "string", "name": "string", "remain_cnt": 0, "price": 0 }

(2) [GET] inventory/get_sandwich_ingredient_inventory/int:page_num/

용도	샌드위치 재료 데이터 SELECT API 한 페이지당 10개씩 출력
테스트 url	http://127.0.0.1:8000/inventory/get_sandwich_ingredient_inventory/1/
테스트 curl	curl -X GET "http://127.0.0.1:8000/inventory/get_sandwich_ingredient_inventory/1/" -H "accept: application/json" -H "X-CSRFToken: ?"
return	{ "detail": [{ "type": "TOPING", "name": "토마토", "price": 500, "remain_cnt": 97 }], "status": 200 }

(3) [POST] inventory/delete_sandwich_ingredient_inventory/

용도	특정 샌드위치 재료 삭제
테스트 url	http://127.0.0.1:8000/inventory/delete_sandwich_ingredient_inventory/
테스트 curl	curl -X POST "http://127.0.0.1:8000/inventory/delete_sandwich_ingredient_inventory/" -H "accept: application/json" -H "Content-Type: application/json" -H "X-CSRFToken: ?" -d '{"type": "치즈", "name": "고다"}'
body	{ "type": "치즈", "name": "고다" }

(4) [GET] inventory/get_sandwich_price/

용도	샌드위치 가격 측정
테스트 url	http://127.0.0.1:8000/inventory/delete_sandwich_ingredient_inventory/
테스트 curl	curl -X GET "http://127.0.0.1:8000/inventory/get_sandwich_price/?bread=치아바타&toping=햄%2C치킨&cheeze=모짜렐라&source=올리브오일" -H "accept: application/json" -H "X-CSRFToken: ?"
return	{ "type": "치즈", "name": "고다" }

(5) [GET] inventory/get_sandwich_order/int:page_num/

용도	샌드위치 주문 내역 출력 (삭제된 데이터는 제외)
테스트 url	http://127.0.0.1:8000/inventory/get_sandwich_order/1/
테스트 curl	curl -X GET "http://127.0.0.1:8000/inventory/get_sandwich_order/1/?search_type=토핑&search_name=치킨" -H "accept: application/json" -H "X-CSRFToken: ?"
params	필터링 시에만 사용 - search_type : 타입 (빵, 토핑, 치즈, 소스) - search_name : 재료명 (치아바타, 토마토, 모짜렐라)
return	{ "status": 200, "detail": [{ "sandwich_no": 39, "ingredient_list": ["치아바타", "토마토", "햄", "고다", "올리브오일"] }] }

(6) [POST] inventory/set_sandwich/

용도	샌드위치 주문 내역 저장
테스트 url	http://127.0.0.1:8000/inventory/set_sandwich/
테스트 curl	curl -X POST "http://127.0.0.1:8000/inventory/set_sandwich/" -H "accept: application/json" -H "Content-Type: application/json" -H "X-CSRFToken: ?" -d '{"bread": "치아바타", "toping": "햄,치킨", "source": "올리브오일", "cheeze": "모짜렐라"}'
body	{ "bread": "치아바타", "toping": "햄,치킨", "source": "올리브오일", "cheeze": "모짜렐라" }
return	{ "status": 200, "detail": { "total_price": 4500 } }

(7) [POST] inventory/delete_sandwich_order/

용도	샌드위치 주문 내역 삭제 is_delete 컬럼 0 → 1 수정
테스트 url	http://127.0.0.1:8000/inventory/delete_sandwich_order/
테스트 curl	curl -X POST "http://127.0.0.1:8000/inventory/delete_sandwich_order/" -H "accept: application/json" -H "Content-Type: application/json" -H "X-CSRFToken: ?" -d '{"sandwich_no": 34}'
body	{ "sandwich_no": 34 }

3.본 기술테스트를 수행한 후의 간단한 소감 및 테스트에 대한 피드백

미니 프로젝트를 진행하며 Django 구조에 대해 다시 복습할 수 있었습니다.

CRUD 전반적인 부분이나 ORM, swagger 등 Django의 기초적인 사용법을 다시 한 번 더 익힐 수 있는 시간이었습니다. 다양한 패키지들을 사용해보며 Django의 장,단점에 대해서도 생각할 수 있었고, Flask 와 비교하며 사용해 볼 수 있어 좋았습니다.

72시간이라는 충분한 시간동안 진행된 과제였고, Django의 전반적인 내용을 다루기에 좋은 프로젝트로 보여 테스트 모두 만족스러웠습니다.