

메뉴선택 - 주문생성 시스템 기능 검증 자동화

1. 과제 요약

1) 과제 목표

- a) 주어진 API 명세에 맞춰 메뉴선택 - 주문생성 시스템의 핵심 기능을 검증하는 테스트 자동화 코드 구현

2) 세부 목표

- a) 명세서 기반 기능 체크리스트 도출
 - i) 정상 케이스
 - ii) 에러 케이스
- b) 체크리스트 기반 TC 작성
- c) 예외 케이스 도출
 - i) 환경 요인
 - ii) 보안
 - iii) 실제 운영에서 발생할 수 있는 오류 상황
- d) 환경에 구애받지 않고 실행 가능하면서, 가벼운 테스트 자동화 설계
- e) 테스트 리포트 생성

2. TC 설계

● 목표

- 핵심 기능에 대해 누락없이 케이스 작성
- API 특성을 고려한 케이스 작성
- 이해하기 쉽게 간결한 작성

1) 분류 정의

a) 기능

i) 정상

ii) 에러

- (1) 인증 · 권한
- (2) 데이터 누락
- (3) 유효성
- (4) 잘못된 데이터
- (5) 에러코드 및 메시지

b) 예외

i) 환경

- (1) 서버
 - (a) SSL 인증서, 서버 다운..
 - (b) 테스트서버 여부

(2) 클라우드

(3) DB

ii) 호환성

- (1) 브라우저
- (2) 디바이스
- (3) 버전

iii) 보안

- (1) 민감정보 노출
- (2) SQL 인젝션
- (3) 세션 · 토큰 만료

iv) 성능

- (1) 응답 시간
- (2) 트래픽
- (3) 대량 호출

2) 기능 체크리스트 작성

기능	Action	상태	체크리스트
헤더	Headers		명세된 Header값이 입력되었는지 확인
			Response가 Content-Type에서 지정한 형식으로 출력되는지 확인
			권한이 유효한지 확인
메뉴예약	Request		모든 파라미터 값을 입력했는지 확인
			quantity값이 1 이상 99 이하의 정수값인지 확인
	Response	SUCCESS	응답 상태 코드가 200 인지 확인
			status값이 'SUCCESS' 인지 확인
			message값이 메뉴 예약 완료 문구인지 확인
			timestamp값이 실제 응답 시간인지 확인
			data Field는 status가 'SUCCESS'인 경우만 출력되는지 확인.
			data.reservationId가 고유한 값인지 확인
			data.reservationId가 예약 만료 시간 이후 비유효한지 확인
			data.reservationExpiresAt 값이 응답 시간 기준 5분 후 값인지 확인
			data.menuId, data.quantity 값이 Request시 입력한 값과 일치하는지 확인
		ERROR	응답 상태 코드 값이 4xx나 5xx 인지 확인
			status값이 'ERROR' 인지 확인

			message값이 Error Code에 대한 설명인지 확인
			errorCode 값이 “INSUFFICIENT_INGREDIENTS”, “INVALID_REQUEST”, “MENU_NOT_FOUND” 중에서 출력되는지 확인
			errorCode값이 응답 상태 코드 값이 4xx 인 경우에만 출력되는지 확인
			timestamp 값이 실제 응답 시간인지 확인
주문생성	Request		메뉴 예약이 완료된 상태인지 확인
			모든 파라미터 값을 입력했는지 확인
			모든 값이 문자형인지 확인
			reservationId값이 memberNo 값으로 생성된 예약ID인지 확인
			memberNo값이 메뉴를 예약한 회원번호인지 확인
	Response	SUCCESS	응답 상태 코드가 200 인지 확인
			status값이 ‘SUCCESS’ 인지 확인
			message값이 메뉴 주문 생성 완료 문구인지 확인
			timestamp값이 실제 응답 시간인지 확인
			data Field값이 status가 ‘SUCCESS’인 경우에만 출력되는지 확인
			data.orderNo값이 고유한 8자리의 영/숫자 값인지 확인 (7 or 9자리 불가)
			data.orderStatus 값이 ‘INITIALIZING’ 인지 확인
			data.reservationId 값이 Request시 입력한 값과 일치하는지 확인

			data.createAt 값이 응답 시간과 동일한지 확인
			data.memberInfo.memberNo 값이 메뉴 예약, Request시 입력한 값과 일치하는지 확인
		ERROR	응답 상태 코드 값이 4xx 나 5xx 인지 확인
			status값이 'ERROR' 인지 확인
			message값이 Error Code에 대한 설명인지 확인
			errorCode 값이 "RESERVATION_EXPIRED", "INGREDIENTS_EXHAUSTED", "INVALID_RESERVATION" 중에서 출력되는지 확인
			errorCode값이 응답 상태 코드 값이 4xx 인 경우에만 출력되는지 확인
			timestamp 값이 실제 응답 시간인지 확인

3. TC 작성

- 체크리스트 기반으로 작성 (엑셀)

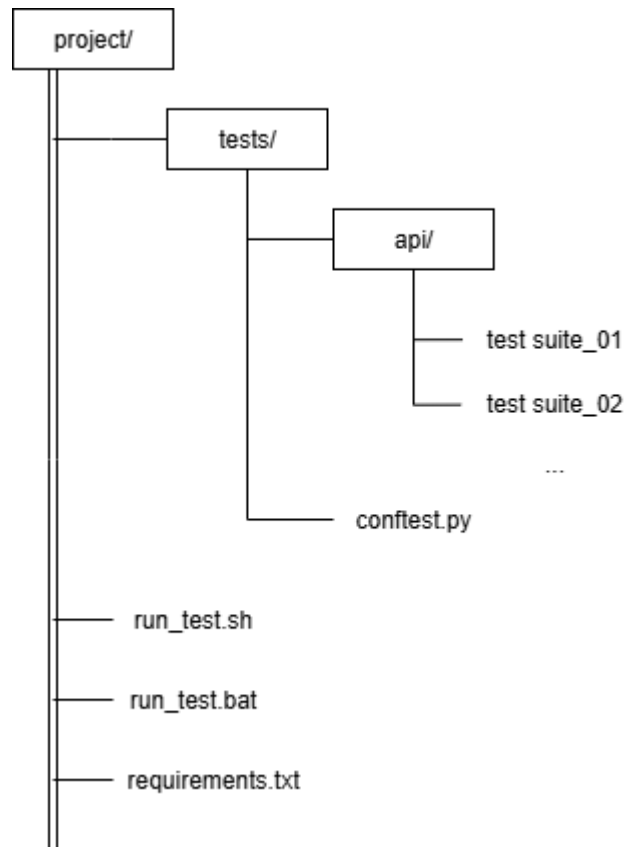
4. 시스템 설계

- 기술 스택 : Python(requests, pytest, pytest-html)
 - requests와 pytest 파이썬 라이브러리를 이용한 코드 작성 및 테스트
 - 사용 이유 : 의존성 면에서 가볍고 간편하게 자동 테스트 구현 가능하며, unittest 대비 빠르게 개발 가능할 것으로 판단
- 환경 세팅
 - Windows

- Python 3.12 설치
- 테스트 관련 라이브러리 설치
- 배포 방식
 1. 실행 파일 생성
 - a. 내부에 python 가상환경 세팅 + requirements 설치 + 테스트 실행문 포함
 - i. Windows : bat 파일
 - ii. Linux/Mac OS : sh 파일
 - b. 테스트 결과... 실행 방식이 간단하고 안정적으로 실행됨
 2. pyinstaller 라이브러리로 실행파일 배포
 - a. 간단하고 직관적이며 Python 설치까지 가능한 장점
 - b. 테스트 결과... 환경에 따라 에러가 많이 발생해서 오히려 불안함
 3. Docker 컨테이너로 배포
 - a. 독립적인 환경으로 일관된 테스트 환경이 보장된다는 장점
 - b. Docker 설치가 필요하며, 내부망이나 보안된 서버에서 접속하는 경우 Docker 네트워크 설정을 별도로 해줘야 함
 - c. 타 사용자의 실행 환경에 대한 정보가 없어 오히려 에러 발생 확률이 높을 것으로 판단..

5. 코드 설계

- a. 아키텍처



- i.
- ii. test case들을 분류별로 묶어 test suite로 관리
- iii. test suite 그룹과 conftest(common 함수), init 등의 파일과 분리