

데이터베이스 통합 테스트 환경 구축하기

Feat. Docker, Jest

박민규 GDSC Soongsil @mingyuu.dev@gmail.com



왜 데이터베이스 통합 테스트가 필요할까?

- In-memory DB의 한계 Production 환경과 Test 환경에서 사용하는 DB가 다름
 Production 환경에서만 실행 가능한 복잡한 쿼리(ex)트리거, 프로시저...) 테스트 불가능
- 시스템이 DB에 의존적인 경우 Unit 테스트를 하기 쉽지 않음



통합 테스트용 DB는 어떤 기능이 필요한가?

- Production 환경과 동일한 스펙을 가진 DB를 사용해야함
- 테스트 실행 시 로컬에 배포 환경과 동일한 DB를 생성하고, 테스트가 종료되면 생성한 데이터를 삭제하여 다음 테스트에 영향을 주지 않게 해야함



도커 쓰면 어떤 점이 좋을까?

- 어디에도 영향을 주지 않는 완전히 격리된 환경에서 실행 할 수 있음
- 테스트 시작할 때 컨테이너 생성하고, 테스트 끝나면 컨테이너 지우면 됨
- docker-compose를 이용하면 테스트 실행 환경까지 쉽게 격리 시킬 수 있음

최종 목표

- 커맨드 하나로 테스트 실행 `npm run test`
 - DB에 스키마가 생성되고, DB 서버가 제대로 뜬 후에 테스트를 실행
- 테스트 결과에 따라 올바르게 exit code 리턴 성공 시 0 리턴
- 테스트 종료 시 컨테이너 제거

Google Developer Student Clubs

Soongsil University

1차 시도 - Fail

```
// docker-compose.yml
version: '3.8'
services:
   image: mysql:latest
   environment:
     MYSQL_ROOT_PASSWORD: root
     MYSQL_DATABASE: test
   ports:
     - '3306:3306'
     - ./mysql-init-files/:/docker-entrypoint-initdb.d/
  app:
   build:
     context: .
     dockerfile: node.dockerfile
   command: npx jest
   environment:
     - DATABASE_URL=mysql://root:root@db:3306/test
   volumes:
     - .:/app
   depends_on:
     - db
   tty: true
```

```
// package.json
{
    "scripts": {
        "test": "docker-compose up",
        "posttest": "docker-compose
do≩n"
}
```

// node.dockerfile
FROM node:lts
WORKDIR /app
COPY . .
RUN npm install

실패 원인

• DB 컨테이너가 올라가고, 포트가 제대로 열리기 전에 app 컨테이너의 Test가 실행 됨

=> APP 컨테이너에서 DB의 컨테이너를 확인하면서 3306번 포트가 열릴 때까지 기다려야함

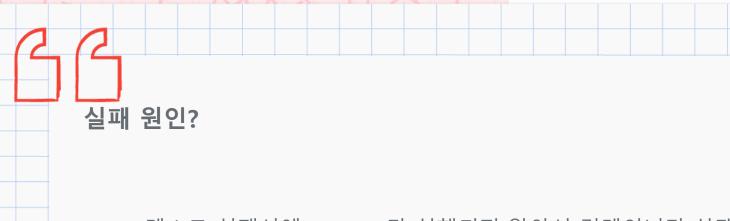
• => <u>https://github.com/vishnubob/wait-for-it</u> 의 쉘스크립트 사용



Soongsil University

2차 시도 - Fail

```
version: '3.8'
services:
  db:
  app:
   build:
     context: .
     dockerfile: node.dockerfile
   command: /bin/bash -c "chmod +x /app/wait-for-it.sh && /app/wait-for-it.sh db:3306 -t 60 -- npx
jest"
   environment:
     - DATABASE_URL=mysql://root:root@db:3306/test
   volumes:
    - .:/app
   depends_on:
     - db
    tty: true
```



• 테스트 실패시에 posttest가 실행되지 않아서 컨테이너가 삭제되지 않음 => 무조건 `docker-compose down`을 실행하고 `docker-compose up`의 결과를 리턴하는 쉘스크립트 작성

진짜 성공 // package.json #!/bin/bash script: { docker-compose up --exit-code-from app "test": "sh ./test.sh" status=\$? docker-compose down exit \$status

