

# Docker를 활용한 코드 테스트 API 구현

최효현\*, 이병선<sup>o</sup>

\*인하공업전문대학 컴퓨터정보과,

<sup>o</sup>인하공업전문대학 컴퓨터정보과

e-mail: hchoi@inhac.ac.kr\*, dlqudtjs922@gmail.com<sup>o</sup>

## Implementation of code test API using Docker

Hyo Hyun Choi\*, Byeong Sun Lee<sup>o</sup>

\*Dept. of Computer Science, Inha Technical College,

<sup>o</sup>Dept. of Computer Science, Inha Technical College

### ● 요약 ●

본 논문에서는 Docker를 활용하여 코드 테스트 API를 활용한 코딩 테스트 웹 사이트를 구현한 결과를 보인다. 코드 테스트에서 다양한 언어 지원을 위해 Dockerfile 템플릿을 개발하여 언어의 다양성을 확보하고 Docker를 이용하여 격리된 환경에서 코드를 실행하고 평가하여 안전하고 효율적으로 채점을 수행한다. Docker의 curl[1]를 이용해 실시간으로 서버와 통신을 통해 결과를 전송한다. 이를 통해 사용자는 작성한 코드에 대한 평가를 실시간으로 피드백 받을 수 있다.

**키워드:** API(API), 도커(Docker), 도커파일(Dockerfile), curl(curl), 코드테스트 (code test)

## I. Introduction

사용자가 작성한 코드의 출력이 정확하게 이루어지는지, 코드의 메모리 사용량 및 실행 시간과 같은 성능을 확인해야 할 경우가 많다. 그러나 이러한 테스트의 결과를 정확하게 얻기 위해 수행되어야 할 작업들이 많다. 기존에 사용하였던 방법은 테스트 값을 직접 입력하여 테스트 결과를 육안으로 확인하고 추가적인 작업을 통해 코드의 세부 사항들을 확인하는 방법이다. 본 논문에서는 이보다 편리한 자동화된 방안을 제안한다. 테스트를 수행하는 코드가 안전하게 작동하며, 다양한 언어를 지원하고, 사용자가 원하는 테스트의 결과 내용을 손쉽게 얻을 수 있다. 사용자가 해당 API를 사용하여 자신이 작성한 코드에 대해 안전하게 테스트할 수 있는 환경을 제공할 수 있으며, 코딩 테스트, 성능 테스트 등 여러 분야에서 사용될 수 있다.

## II. Preliminaries

### 1. 사례

국민대학교에서 유사한 방식으로 알고리즘 채점 서버를 구축하였다.[2] 채점할 사용자의 코드 각각을 개별적인 docker container 안에서 처리하고 있음을 알 수 있다. 이러한 방식을 통해 안전하고 격리된 환경에서 채점을 수행할 수 있었고, 동시에 다양한 사용자들의 코드를 독립적으로 실행하며 성능과 보안 측면에서 우수한 결과를

얻을 수 있었다. 알고리즘 채점 사이트뿐만 아니라 웹 컴파일러를 유사한 방식으로 구현한 프로젝트 또한 있으며[3] 이는 N명의 사용자가 각각 다른 환경에서 실행할 수 있도록 Docker 컨테이너 안에서 컴파일을 수행하여 사용자에게 결과를 보여준다. 본 논문에서는 사례에서 보았듯이 다양한 환경에서 사용될 수 있는 코드 테스트 API를 구현하여 제공하는 방안에 중점을 두었다.

## III. Development

### 1. 전체 구조

이 프로젝트는 코드 테스트 API를 사용하여 코딩 테스트 웹 사이트 구현에 중점을 맞춘 프로젝트이기 때문에 사용자의 코드를 작성하여 서버에 전송하는 클라이언트, MVC, Rest API[4]로 구현된 서버, 사용자의 코드를 채점하는 API가 논리적으로 분리되어 있다.

클라이언트는 작성된 코드와 언어, 코딩 테스트 문제를 선택하여 서버에 전송하게 되면 서버는 문제에 맞는 테스트 입력과 출력을 데이터베이스에서 가져오게 된다. 그 후 서버는 사용자의 코드 및 테스트 입력과 정답 결과를 채점 API에 전달하게 되고, 채점 API는 Dockerfile 템플릿을 통해 사용자 코드를 컴파일할 수 있는 컴파일러를 설치하고, 사용자 코드를 Script[5] 형식에 맞게 컴파일한다. 또한,

Dockerfile에서 컴파일된 프로그램에 테스트 케이스를 입력하고 출력 결과와 정답을 비교하며 코드의 실행 시간과 메모리 사용량을 측정하여 서버에게 전송하는 역할을 하는 Script를 Docker Container 내부에서 실행하게 되며 결과를 전송받은 서버는 결과값을 데이터베이스에 저장한 후 클라이언트에게 결과값을 전송하여 최종적으로 사용자가 작성한 알고리즘의 정답 유무를 확인할 수 있다.

## REFERENCES

- [1] <https://hub.docker.com/r/curlimages/curl>
- [2] <https://algotab.kookmin.ac.kr/manual>
- [3] [https://github.com/DongGeon0908/WebCompiler\\_SE.LAB](https://github.com/DongGeon0908/WebCompiler_SE.LAB)
- [4] <https://ko.wikipedia.org/wiki/REST>
- [5] [https://en.wikipedia.org/wiki/Bash\\_\(Unix\\_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))
- [6] <https://github.com/dlqudtjs/montshell>

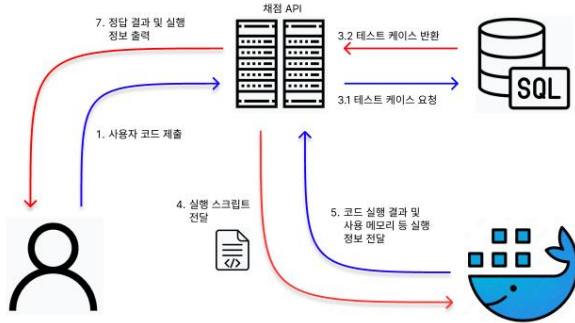


Fig. 1. Process Flow

## IV. Result

Docker를 활용한 코드 테스트 API를 활용하여 독립적으로 사용자의 코드를 채점하여 정답 유무와 메모리 사용량 및 코드의 실행 시간 등 여러 정보를 웹 사이트에서 확인할 수 있으며 해당 프로젝트는 Github에 게시되어 있다.[6]

제출 번호	아이디	문제	결과	메모리	시간	사용 언어	제출시간
23	user	2	Wrong	36032 KB	35ms	java	2023-05-21 15:38
24	user	2	Wrong	38072 KB	33ms	java	2023-05-21 15:50
25	user	2	Wrong	48712 KB	56ms	java	2023-05-21 15:57
26	user	2	Correct	46588 KB	50ms	java	2023-05-21 16:01
27	user	2	Correct	50488 KB	62ms	java	2023-05-21 16:08
28	user	2	Wrong	213824 KB	816ms	java	2023-05-21 16:17
29	user	2	Correct	217136 KB	881ms	java	2023-05-21 16:20
30	user	1	Correct	39276 KB	41ms	java	2023-05-24 02:56
31	user	1	Correct	36816 KB	32ms	java	2023-05-24 03:08
35	user	1	Correct	41056 KB	34ms	java	2023-05-17 19:02

Fig. 2. Scoring Results Page

## V. Conclusions

본 논문에서는 기존의 방식보다 더욱 편리하고 안전하게 코드를 테스트할 수 있는 코드 테스트 API를 활용하여 만든 코딩 테스트 웹 사이트를 구축하였고, 사용자에게 채점 결과와 코드에 대한 상세 정보를 보여주었다. 또한, 사례에서 살펴보았듯이 코딩 테스트 웹 사이트뿐만 아니라 코드의 성능 테스트, 예외 테스트, 웹 컴파일러 구축 등 여러 테스트 분야에서 사용할 수 있을 것이다. 추후에는 코드 테스트 API를 활용해 코드의 알고리즘의 복잡도를 분석하는 개발에 집중할 계획이다.