

Cypress로 E2E 테스트 시작하기

실무에 바로 적용하는 프론트엔드 테스트



- ✓ 자바스크립트를 사용해 작성
- ✓ 실제 웹 앱에서 다양한 테스트를 작성할 수 있는 오픈 소스 자동화 도구

<https://www.cypress.io/>

Cypress의 특징

- ✓ Node.js 서버 프로세스를 통해 변경이 발생할 경우 실시간으로 동기화
- ⚠ 테스트 실행 중 노드 서버를 통해 파일 시스템에 접근 하는 것도 가능

Cypress의 특징

- ✓ Node.js 서버 프로세스를 통해 변경이 발생할 경우 실시간으로 동기화
- ✓ OSI Model 네트워크 계층에서 웹 트래픽을 즉시 읽고 변경할 수 있음
 - ➡ 브라우저에 들어오는 요청을 변경할 수 있음
- 💪 브라우저 안·밖 상황을 제어해 일관된 환경을 제공할 수 있음

Cypress의 특징

- ✓ Node.js 서버 프로세스를 통해 변경이 발생할 경우 실시간으로 동기화
- ✓ OSI Model 네트워크 계층에서 웹 트래픽을 즉시 읽고 변경할 수 있음
 - ✓ E2E 테스트 뿐만 아니라 컴포넌트 테스트도 제공
- 😊 강의에서는 vitest 환경에서 단위·통합 테스트를 검증하고 있기 때문에 cypress는 E2E 테스트 목적으로만 사용

사용 브랜치



shopping-mall-e2e-test-example-with-answer

```
$ git checkout shopping-mall-e2e-test-example-with-answer
```

Cypress 설치 최소 사양

- ✓ Windows 10, macOS 10.9, Linux Ubuntu 20.04, Fedora 21,.. ↑
 - ✓ Node.js 18.x, 20.x ↑
 - ✓ 최소 4GB 이상의 메모리

⚠ 사양은 사용 버전에 따라 다름. 확인은 아래 링크에서!

<https://docs.cypress.io/guides/getting-started/installing-cypress#System-requirements>

Cypress 설치

```
$ npm i -D cypress
```



출처: <https://docs.cypress.io/guides/getting-started/installing-cypress>

테스트의 대상이 되는 서버

테스트 서버를 대상으로 테스트를 작성할 경우

😊 실제 환경과 유사한 환경에서 테스트 → 안정성 향상

🤔 테스트 환경에 배포 후에 테스트 실행 가능하며, 유지 보수 증가

로컬 서버를 대상으로 테스트를 작성할 경우

😊 배포 전 미리 검증할 수 있음. 테스트 환경 및 유지 보수가 훨씬 간편해짐

🤔 테스트 결과가 실제 서비스의 안정성을 의미하지는 않음

head 모드와 headless 모드

head 모드

- ✓ cypress open
- ✓ 브라우저 UI까지 모두 구동
- ✓ 시각적으로 확인 가능하기 때문에 디버깅 시 많이 이용

headless 모드

- ✓ cypress run
- ✓ 브라우저 UI 없이 브라우저 엔진을 명령어 인터페이스로 제어
- ✓ 구동 속도가 상대적으로 빨라 CI 또는 클라우드 환경에서 구동하기 적합

정리

Cypress

- 실제 웹 앱을 기준으로 다양한 테스트를 작성할 수 있는 오픈 소스 자동화 도구
- 브라우저 안과 밖에서 일어나는 상황을 제어할 수 있어 일관된 환경에서 테스트를 실행할 수 있다

Head모드와 Headless모드

- head 모드
 - 브라우저 UI까지 모두 구동하여 시각적으로 확인할 수 있는 환경에서 테스트 실행
 - 주로 실행 과정을 확인하거나 디버깅 시 사용
- headless 모드
 - UI없이 브라우저 엔진을 명령어 인터페이스로 제어하여 테스트를 실행
 - 구동속도가 상대적으로 빨라 CI 또는 클라우드 환경에서 사용

Cypress의 장점

- 편리하고 빠른 디버깅
- Time Travel과 스크린샷