

TDD Study 1주차

Node.js 기반의 REST API 서버 개발

Express.js

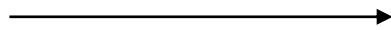
- 웹 프레임워크
- 요청에 대한 응답을 편하게 처리

/home 링크



Home page

/user 링크



User page

‘라우팅’

**어플리케이션으로 뼈대를 만들고
미들웨어로 함수를 추가하자**

어플리케이션으로 뼈대를 만들고 미들웨어로 함수를 추가하자



일반 미들웨어
에러 미들웨어

REST API

REpresentational **S**tate **T**ransfer

웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처

키워드1 Resource

- 명사로 나타낼 수 있는 어떠한 정보 (학생, 사용자 등)

키워드2 Method

- 동사로 나타낼 수 있는 변화 수행에 사용 (검색한다 등)

REST API

GET localhost:3000/users

행동은 메소드(동사)로 표현

리소스는 명사로 표현

REST API

GET localhost:3000/users/1



리소스에 대한 식별자
(일반적으로 id)

TDD에 발 담그기

mocha, should, supertest

npm install <package> --save-dev

Mocha

테스트 코드를 돌려준다.

→ 테스트 꾸러미(환경)

```
describe('어떤 테스트를 하겠다', () => {
```

```
  it('이렇게 실행되어야 한다', (done) => {
```

```
    /* 테스트 코드 */
```

```
  });
```

→ 테스트 케이스(실제 테스트)

```
});
```

→ 테스트 종료를 알리는 함수

Should

검증(assertion) 라이브러리

가독성 높은 테스트 코드 작성 가능 (영어와 유사)

Supertest

통합 테스트용 라이브러리

내부적으로 express 서버를 구동하여 실제 요청을 보낼 수 있음

