CLIENT 요청

with request.params, request.body, rquest.query

AWS 인바운드
보안규칙

PORT: 8000

서버님~ 요청(req)에 응답(res)좀 주세요.

req.params
req.body
~~req.query~~ 가져왔음!

AWS-EC2
ubuntu

틀

POST api/signup
POST api/login
POST api/posts
GET api/posts
GET api/posts/:postId
PUT api/posts/:postId
DELETE api/posts/:postId
PUT api/posts/:postId/like
GET api/posts/like
POST api/comments/:postId
GET api/comments/:postId
PUT api/comments/:commentId
DELETE api/comments/:postId

API#1. 회원가입 | API#2. 로그인 | API#3. 글등록 | API#4. 글조회 | API#5. 상세조회 | API#6. 글수정 | API#7. 글삭제 | API#8. 글좋아요 | API#9. 내좋아요조회 | API#10. 댓글등록 | API#11. 댓글조회 | API#12. 댓글수정 | API#13. 댓글삭제

인스턴스

is Listening Requests....

server.js

app.js

index.js

users.route.js — router
~~posts.route.js~~ — router
comments.route.js — router

auth-middleware.js

Auth 클래스의 인스턴스 auth를 만들어 그 안의 메소드 authMiddleware를 유저인증에 활용

유효토큰인지 인증

users.contr~.js 인스턴스/메소드: method method
posts.controller.js PostController 클래스의 인스턴스/메소드: method ×7
comments.controller.js 인스턴스/메소드: method ×4

validation

users.service.js 인스턴스/메소드: method method
posts.service.js PostService 클래스의 인스턴스/메소드: method ×7
comments.service.js 인스턴스/메소드: method ×4

users.repository.js UserRepository의 인스턴스/메소드: method ×8
posts.repository.js PostRepository 클래스의 인스턴스/메소드: method ×9
comments.repository.js 인스턴스/메소드: method ×5

유효토큰 발행

models/users.js User model의 인스턴스/메소드: 각종 sequelize methods
models/posts.js Post 모델의 인스턴스/메소드: 각종 sequelize methods
models/comments.js 인스턴스/메소드: 각종 sequelize methods

= new UserRepository()

Express 객체와 Router로 기초적인 연결을 형성

Create
Read
Update
Delete

(객체와 DB 테이블의 매핑 모델)
Object - Relational Mapping Model)

index.js — models/
.....post
.....comment
.....user — migrations/
config.json — config/

모델을 정의, 관계를 설정

초기 DB관련
sequelize 환경 설정

sequelize
+sequelize-cli

npx 명령어

mysql2

pm2
Don't Sleep!

변환된 SQL Query

(사실 내부적으로는
별도의 가상환경(cluster)를
만들어서 돌고 있음)

joi | jsonwebtoken | express

npm (node package manager)

Node.js

⋮

Ubuntu OS 18.04

SQL Query

AWS-RDS
mySQL

github

git clone ~

git pull ~

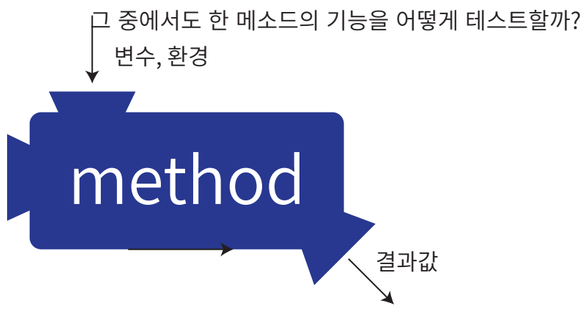
git push ~

내 PC

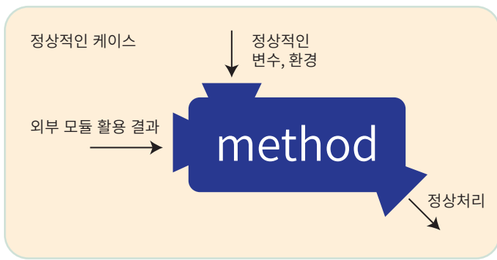
PORT 22

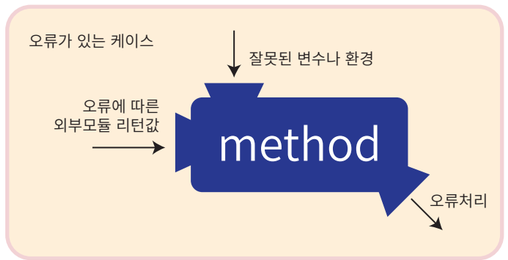
ssh/
원격접속 제어

users.controller.js
UsersController 클래스가 있다.



여러 시나리오(Case)에서 이 단위 자체가 독립적으로 잘 구동하려면
케이스별로 이를 둘러싼 인풋(변수, 환경, 외부모듈 활용한 결과값)을 통제할 수 있어야 한다.





ex. 가짜 reqeust를 보낼 수 있다.

super-test

ex. 가짜 res, req, next
객체를 만들 수 있다.

node-mocks-http

ex. jest.fn()을 통해 가짜 함수로 만들어
메소드 중간에 리턴할 값을 미리 통제할 수 있다.

jest

새로 install 한 이 모듈들은 테스트를 위한 이런 가짜 환경, 즉 mock환경을 만들어내는 도구들이 된다.

테스트코드는 이런식이다.

```
Run | Debug
it("기능: 사용자가 지금까지 좋아한 리스트에 이번 게시글이 없는 경우, ~~~ 이래야 한다.", async () => {
  // 전제된 경우, Repository들이 아래값을 리턴한다고 치자.
  postsService.postRepository.getPost.mockReturnValue(
    postDataOut.getPostDetailRes.data
  );
  postsService.userRepository.getAllLikedPosts.mockReturnValue(["1", "3"]); // 기존에 좋아한 배열이 1,3인 상태에서

  // 로그인한 유저가
  await postsService.likePost(res.locals.user, "4"); // 4번을 좋아하면

  // post저장소와 user저장소의 좋아요(likePost) 메소드가 각각 1번씩 호출되어야 한다.
  expect(postsService.postRepository.likePost.mock.calls.length).toBe(1);
  expect(postsService.userRepository.likePost.mock.calls.length).toBe(1);
});
```

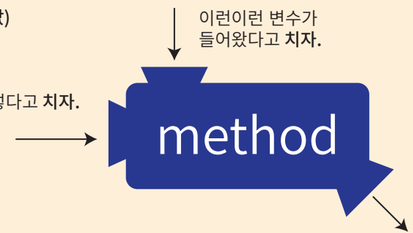
단위 테스트 코드는 각 부품이 해결해야할 **상황을 디자인하고 통제해서 돌려보는 것**이다.

전달 가능한 다양한 케이스의
각종 변수와 환경, 정보

안에서 사용할 외부 메소드
.mockReturnValue(리턴값)

그런 상황에서 당연히
내부메소드들은 이런 값을
정상적으로 리턴하겠지? 그렇다고 **치자.**

이런이런 변수가
들어왔다고 **치자.**



그럼 이런 상황에서 이 메소드는
이럴 것이라고 기대(expect)하는데,
정말 그런가?