

Back-End STUDY

프로젝트에서 1인분 하기

2018. 11. 03

백엔드팀 조현우



CONTENTS

01 역할

02 선택

- Web Server
- Web Application Framework
- Database

03 설계

- API 설계
- Database 설계

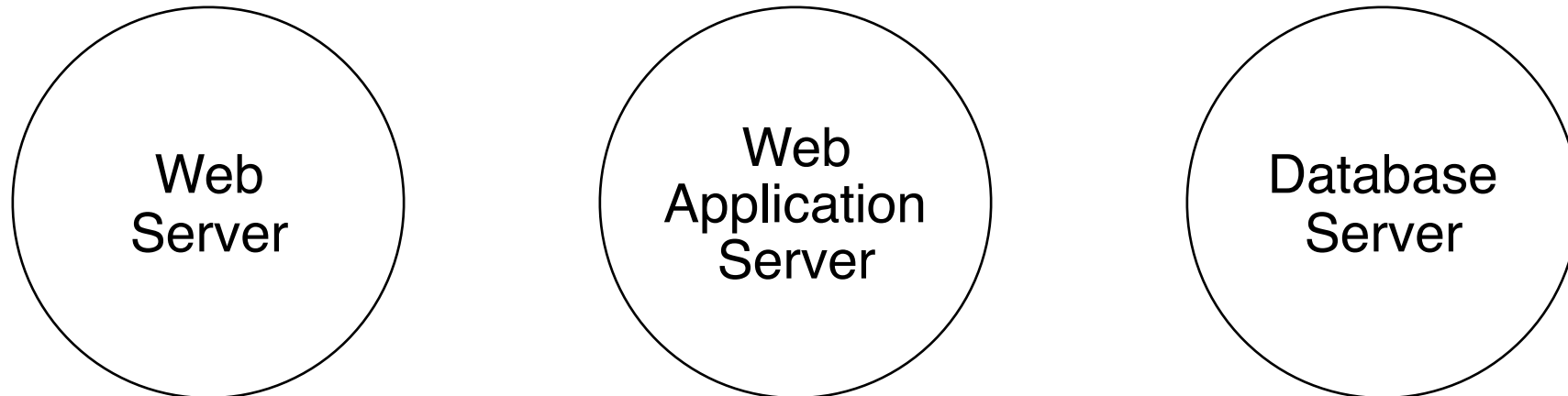
04 배포

- 클라우드 서버
- CI / CD

01

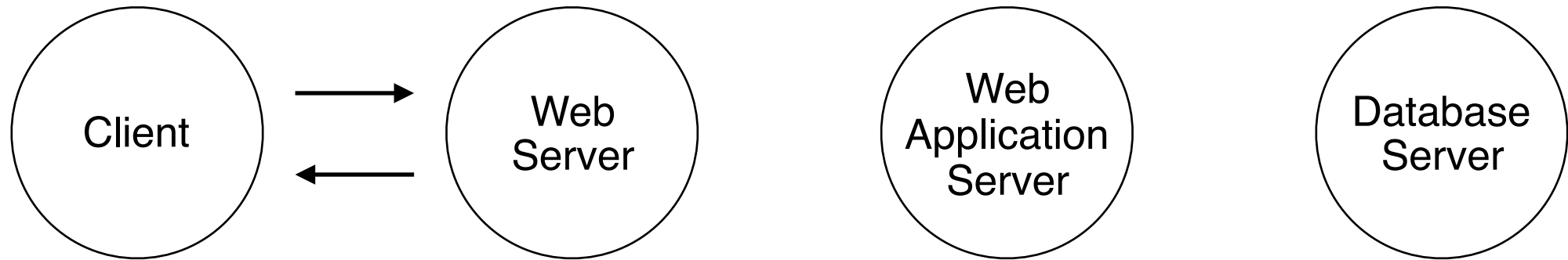
역할

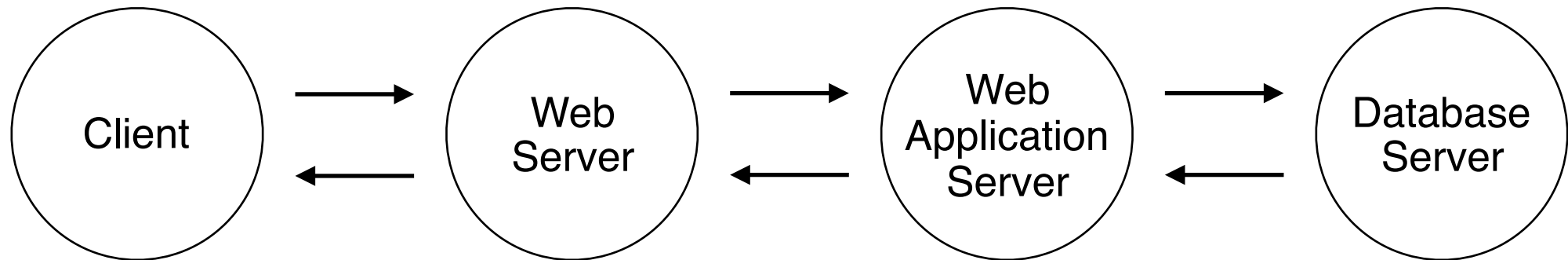
서버 구축

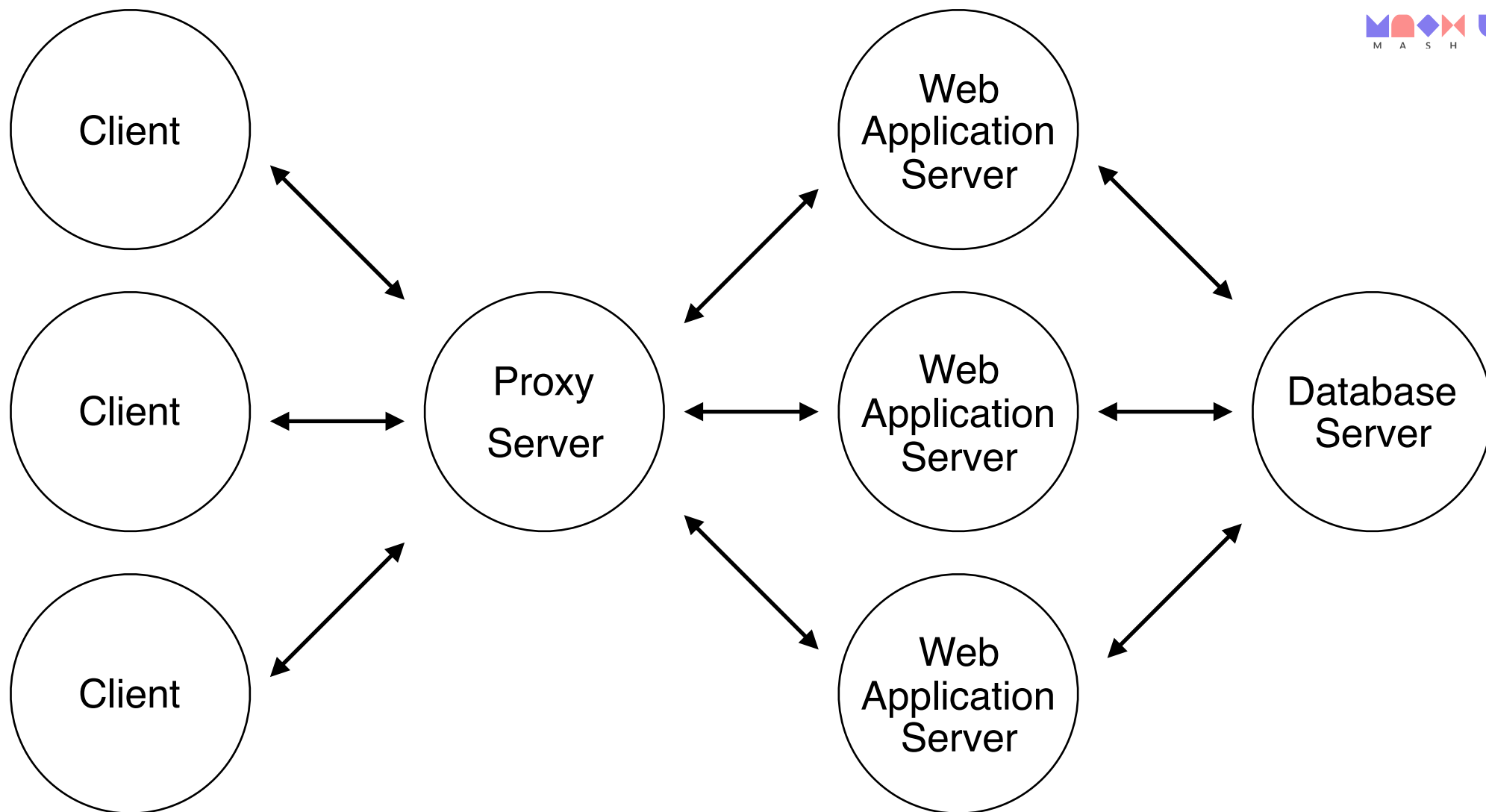


Web Server ?

Web Application Server ?



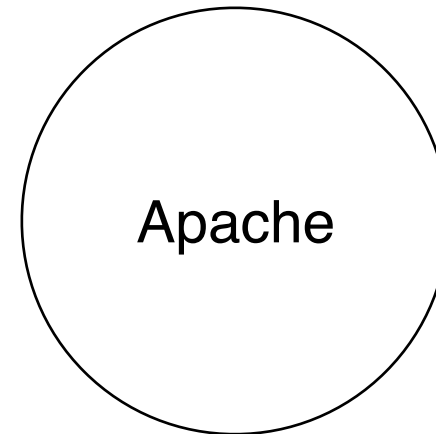
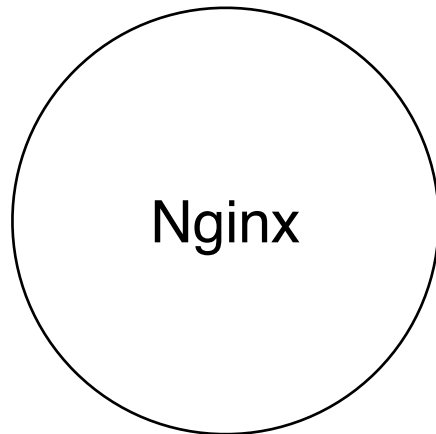




02

선택

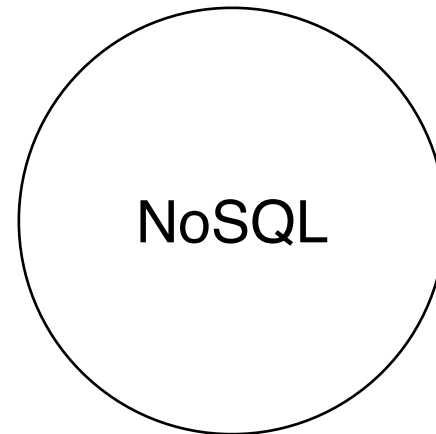
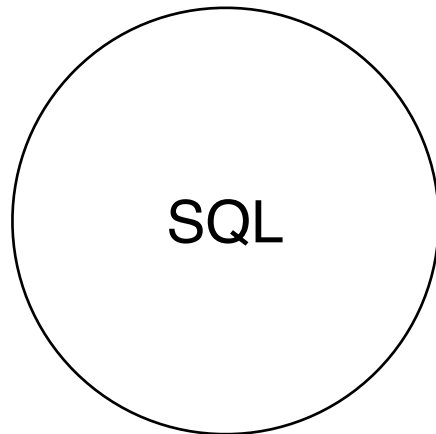
- Web Server
- Web Application Framework
- Database



Django

Express

Spring



SQL

- 규칙에 맞는 데이터 입력
- 테이블 간 JOIN 지원
- 트랜잭션 지원
- 안정성, 일관성
- 용어(테이블, 로우, 컬럼)

NoSQL

- 자유로운 데이터 입력
- 컬렉션 간 JOIN 미지원
- 트랜잭션 미지원
- 확장성, 가용성
- 용어(컬렉션, 다큐먼트, 필드)

03

설계

- API 설계
- Database 설계

REST API

REST API

- URI는 정보의 자원을 표현해야 한다.
- 자원에 대한 행위는 HTTP Method로 표현한다.

REST API

리소스	POST	GET	PUT	DELETE
/customers	새 고객 만들기	모든 고객 검색	고객 대량 업데이트	모든 고객 제거
/customers/1	오류	고객 1에 대한 세부 정보 검색	고객 1이 있는 경우 고객 1의 세부 정보 업데이트	고객 1 제거
/customers/1/orders	고객 1에 대한 새 주문 만들기	고객 1에 대한 모든 주문 검색	고객 1의 주문 대량 업데이트	고객 1의 주문 제거

REST API

- HTTP 상태 코드로 에러 핸들링
- API 버전 관리
- 결과반환 리스트에 필터링, 정렬, 결과요소 선택, 페이징 기능을 제공하기
- etc...

API 문서화

API Endpoint 정리

bigstark edited this page on 29 Sep · 8 revisions

BASE URL

- <http://220.230.121.118:8080> (언제든지 바뀔 수 있음)
- swagger 문서 : <http://220.230.121.118:8080/api/swagger-ui.html>

Auth

Method	endpoint
POST	/api/auth/register
POST	/api/oauth/token

User

Method	endpoint
GET	/api/user/{userId}
POST	/api/user/{userId}/follow
DELETE	/api/user/{userId}/follow
GET	/api/user/{userId}/stamps?count={count}&cursor={curosr}

Me

Method	endpoint
GET	/api/me
PUT	/api/me
GET	/api/me/stamps?count={count}&cursor={curosr}
GET	/api/me/wishlist?count={count}&cursor={curosr}
DELETE	/api/me/wishlist/{wishId}

Review

Method	endpoint
GET	/api/review/{reviewId}
POST	/api/review/{reviewId}/wish

Register

Request

POST /api/auth/register

Header

Key	Value
Content-Type	application/json

Body

Key	Type	Essential	Info
username	string	true	username. 겹치지 않는 값
password	string	true	password

ex)

```
{  "username": "daekyu",  "password": "1234"}
```

Response

Success: 201 (Created)

Fail

Key	Type	Info
code	int	상태 코드
message	string	에러 메시지

ex)

```
{  "code": 400,  "message": "Please input the username"}
```

Fail 이 될 수 있는 경우

400 : BAD_REQUEST

- Essential 한 parameter 가 없을 경우
- username 이 중복될 경우

500 : INTERNAL SERVER ERROR

+ API 문서화 - Swagger



swagger Select a spec default

Api Documentation ^{1.0}

[Base URL: 220.230.121.118:8080/ap1]
<http://220.230.121.118:8080/ap1/v2/api-docs>

Api Documentation
[Terms of service](#)
[Apache 2.0](#)

auth-controller Auth Controller

POST /auth/register register

me-controller Me Controller

GET /me getMe

PUT /me putMe

GET /me/stamps getStamps

GET /me/wishlist getWishes

newsfeed-controller Newsfeed Controller

GET /newsfeed getStamps

auth-controller Auth Controller

POST /auth/register register

Parameters

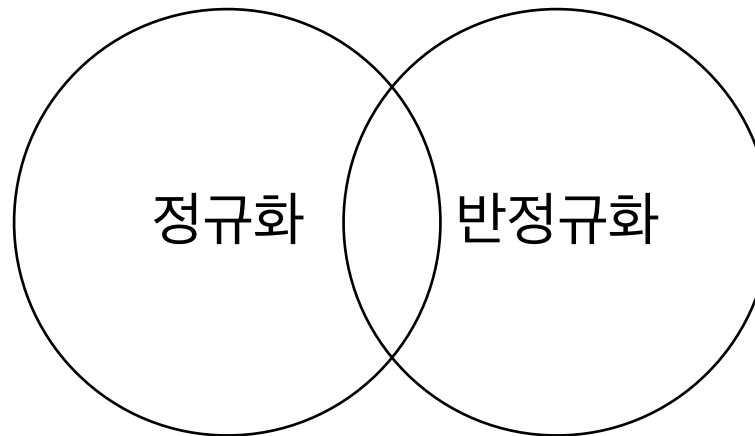
Name	Description
command ^{required} (body)	command Example Value Model <pre>{ "password": "string", "username": "string" }</pre> Parameter content type application/json

Responses

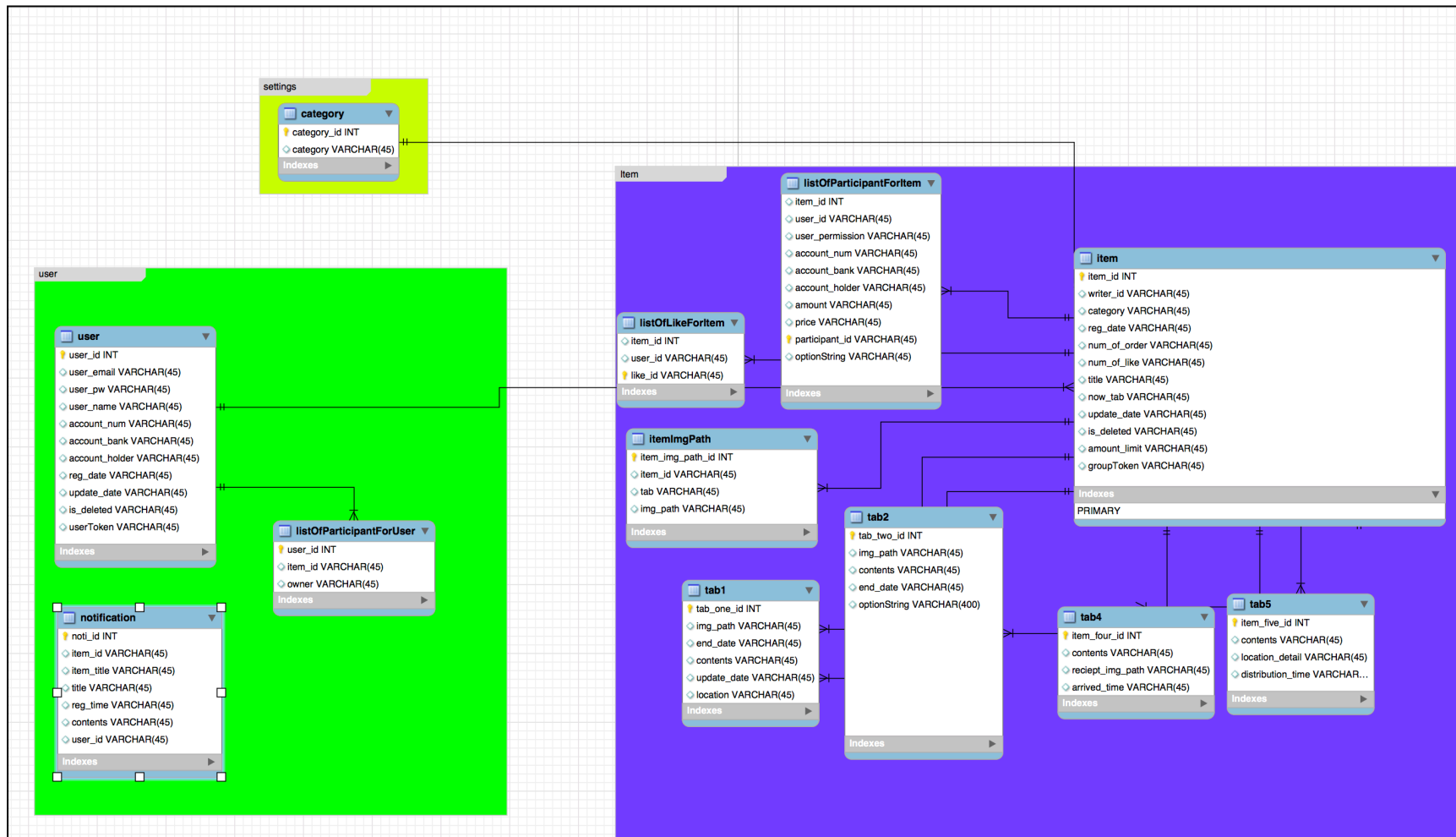
Code	Description
200	OK
201	Created
400	Bad Request
401	Unauthorized
409	Already Exists
500	Failure



RDB



+ ERD(Entity Relationship Diagram)



04

배포

- 클라우드 서버
- CI / CD

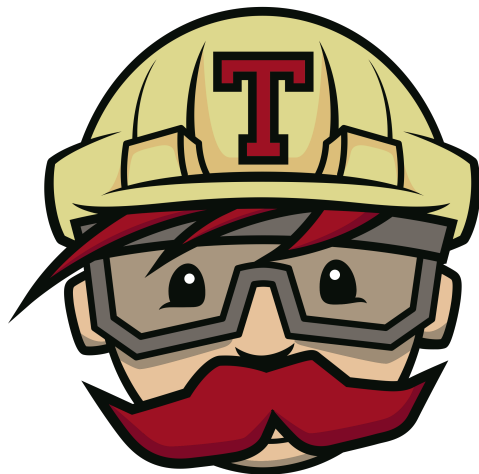


Google Cloud Platform

CI / CD

- CI(Continuous Intergration)
- CD(Continuous Delivery or Continuos Deploy)

빌드, 테스트, 배포 등을 자동으로 해주는 서비스



Travis CI



Jenkins

+ 참고 자료



1. Web Server? Web Application Server?

- <https://okky.kr/article/243427>
- <http://jeong-pro.tistory.com/84>
- <http://limmmee.tistory.com/4>

2. Proxy server

- <http://soul0.tistory.com/230>
- <http://bcho.tistory.com/tag/reverse%20proxy>

3. 로드 밸런서

- <https://nesoy.github.io/articles/2018-06/Load-Balancer>

4. Apache? Nginx?

- <https://m.blog.naver.com/tmondev/220737182315>

5. Django? Node.js? Spring?

- <https://stackshare.io/stackups/django-vs-nodejs-vs-spring>

6. SQL? NoSQL?

- <https://brunch.co.kr/@kooslab/181>

7. REST API

- 개념 : <http://bcho.tistory.com/953>
- 디자인 : <http://bcho.tistory.com/954>
- 보안 : <http://bcho.tistory.com/955>

8. API 문서화

- Github Wiki 참고 : <https://github.com/mash-up-kr/gourmet-backend/wiki/API-Endpoint-%EC%A0%95%EB%A6%AC>
- Swagger? : <https://www.linkedin.com/pulse/swagger-api-framework-12-%EA%B0%80%EC%A7%80-%EC%95%8C%EC%95%84%EC%95%BC%ED%95%98%EB%8A%94-%EA%B2%83%EB%93%A4-eric-jins>
- Swagger 자동화

1. Django : <https://medium.com/towncompany-engineering/%EC%B9%9C%EC%A0%88%ED%95%98%EA%B2%8C-django-rest-framework-api-%EB%AC%B8%EC%84%9C-%EC%9E%90%EB%8F%99%ED%99%94%ED%95%98%EA%B8%B0-drf-yasg-c835269714fc>
2. Express : <https://gongzza.github.io/javascript/nodejs/swagger-node-express/>
3. Spring : <https://ahea.wordpress.com/2017/01/18/spring-boot-swagger-%EC%A0%81%EC%9A%A9%EA%B8%B0/>

9. API Testing 도구

- Postman : <https://www.getpostman.com/>
- Insomnia : <https://insomnia.rest/>

10. CI / CD

- <http://asfirstalways.tistory.com/304>
- <http://woowabros.github.io/experience/2018/06/26/bros-cicd.html>

THANKS FOR WATCHING

감사합니다

2018. 11. 03

백엔드팀 조현우

