

알엠소프트 코딩 테스트

개발 환경

IDE : 인텔리제이
Language: Java17
Framework: SpringBoot3.1.5, JPA
데이터베이스: MySQL8.0, Redis
DevOps: Docker, AWS(Ubuntu 22.04), GithubAction
그 외: Postman API 문서화

Github주소: <https://github.com/brightdev-bs/book-management>

문서화 주소:

<https://documenter.getpostman.com/view/19678580/2s9YeLZpeh#b4bd8b53-97b6-4ba9-92e0-3fde1d06231e>

- **Example Request** 카테고리의 회원가입 화살표를 누르면 각 케이스 별 요청과 응답을 확인할 수 있습니다.

The screenshot displays a Postman interface for a POST request. The URL is `ec2-3-35-21-80.ap-northeast-2.compute.amazonaws.com:8080/members/sign-up`. The request body is in JSON format, containing `{ "email": "test2@gmail.com", "name": "techbear2" }`. Below the body, there is an "Example Request" section, which is highlighted with a red border. This section contains a curl command: `curl --location 'ec2-3-35-21-80.ap-northeast-2.compute.amazonaws.com:8080/membe' --data-raw '{ "email": "test1@gmail.com", "name": "techbear2" }'`. A "View More" button is visible at the bottom right of the curl command area.

POST 회원가입

ec2-3-35-21-80.ap-northeast-2.compute.amazonaws.com:8080/members/sign-up

Body raw (json)

json

```
{
  "email": "test2@gmail.com",
  "name": "techbear2"
}
```

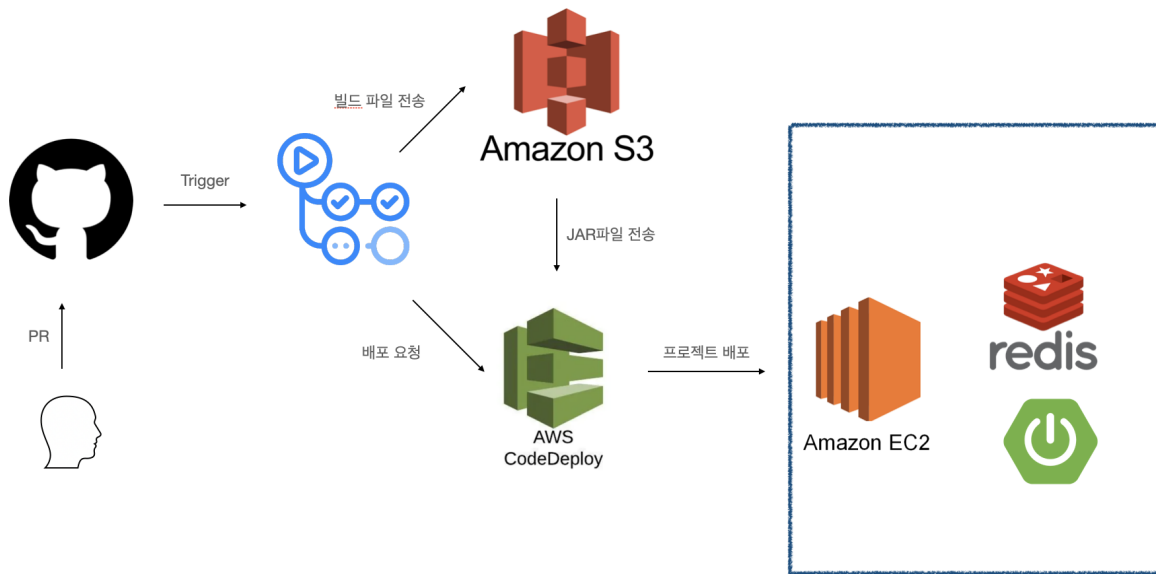
Example Request 회원가입 ▾

curl

```
curl --location 'ec2-3-35-21-80.ap-northeast-2.compute.amazonaws.com:8080/membe' --data-raw '{
  "email": "test1@gmail.com",
  "name": "techbear2"
}'
```

View More

프로젝트 아키텍처



1. [main] 브랜치로 PR을 보내면 Github가 이를 인식하고 Github Action을 트리거 시킨다.
2. Github Action에 정의된 JOB을 실행한다.
 - 프로젝트 파일 빌드
 - S3에 업로드
 - Code-Deploy를 이용한 프로젝트 배포

프로젝트 특징

1. TDD 기반의 프로젝트: 총 27개의 테스트 작성
 - a. 단위 테스트: 17개
 - b. 통합 테스트: 10개
2. Redis를 이용한 연체자 관리
3. AOP를 활용한 예외 처리 및 커스텀 예외 정의
4. 로컬에서도 쉽게 구동할 수 있도록 Docker-compose 작성했습니다.

프로젝트 주소

ec2-3-35-21-80.ap-northeast-2.compute.amazonaws.com:8080

- inbound로 8080번 포트만 열어놨습니다.

Protocol	Port range	Source
TCP	8080	::/0
TCP	22	221.147.245.89/32
TCP	8080	0.0.0.0/0

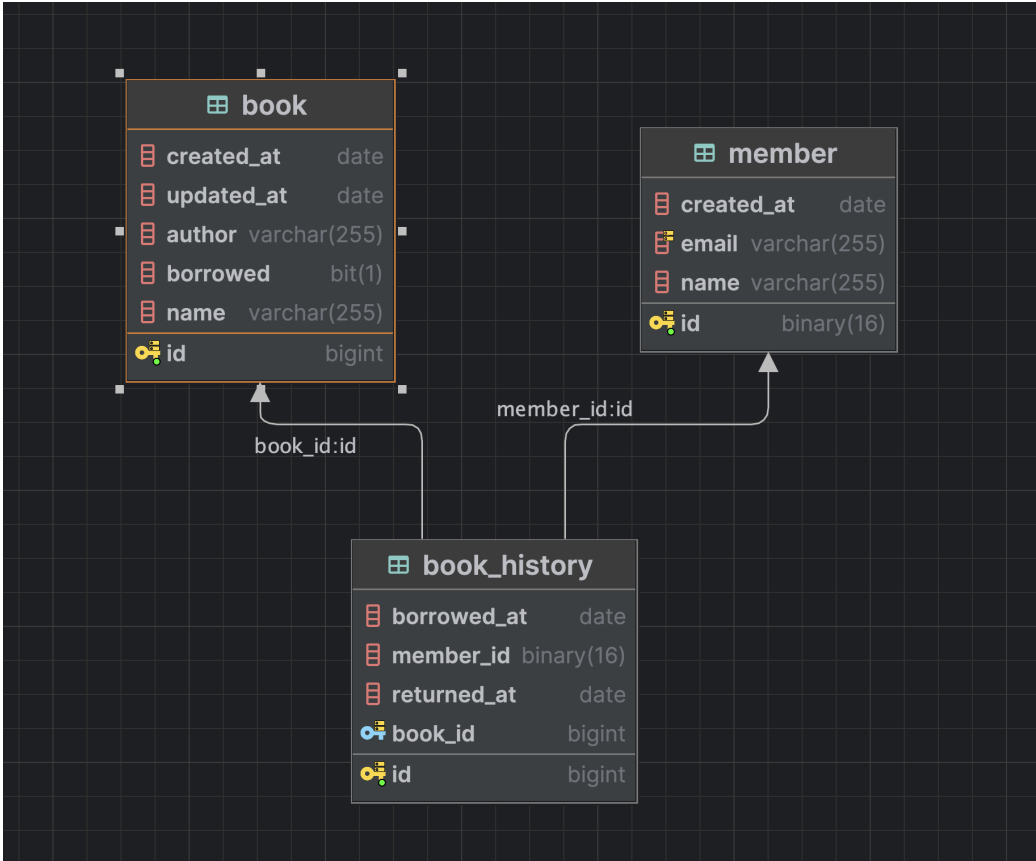
IAM (Identity and Access Management)는 필요한 기능만 선택하여 역할을 부여하였습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "codedeploy:*",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

Code-Deploy에 대한 이력도 확인하고 있습니다.

Deployment history							
<div> <input type="button" value="View details"/> <input type="button" value="Actions"/> <input type="button" value="Copy deployment"/> <input type="button" value="Retry deployment"/> </div> <div> <input type="text"/> <div> < 1 > </div> </div>							
	Deployment Id	Status	Deploym...	Compute ...	Application	Deploym...	Rev
	d-H3O5EC0YE	Succeeded	In-place	EC2/On-p...	book-man...	book-group	s3:
	d-6ATFWW7A2	Succeeded	In-place	EC2/On-p...	book-man...	book-group	s3:
	d-S01L6U6A2	Succeeded	In-place	EC2/On-p...	book-man...	book-group	s3:
	d-Z7SZDU6A2	Succeeded	In-place	EC2/On-p...	book-man...	book-group	s3:
	d-UTMDIOZXE	Succeeded	In-place	EC2/On-p...	book-man...	book-group	s3:

[ERD]



[테이블 정의서]

Book						
컬럼명	id	created_at	updated_at	author	borrowed	name
데이터 타입	bigint	DATE_TIME	DATE_TIME	varchar(255)	boolean	varchar(255)
설명	아이디	등록 날짜	수정 날짜	작가	빌린 상태	책 이름
BookHistory						
컬럼명	id	borrowed_at	member_id	returned_at	book_id (FK)	
데이터 타입	bigint	DATE_TIME	DATE_TIME	varchar(255)	bigint	
설명	아이디	빌린 날짜	회원 아이디	반납 날짜	책 번호	
Member						
컬럼명	id	email (unique)	name			
데이터 타입	bigint (UUID)	varchar(255)	varchar(255)			
설명	아이디	이메일	이름			

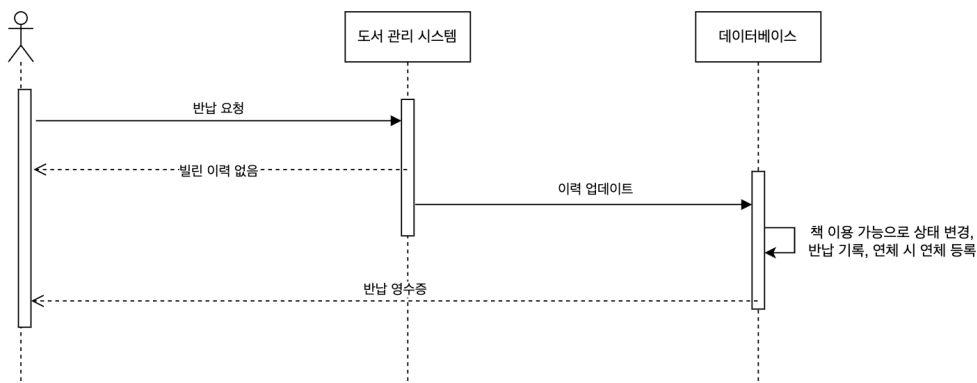
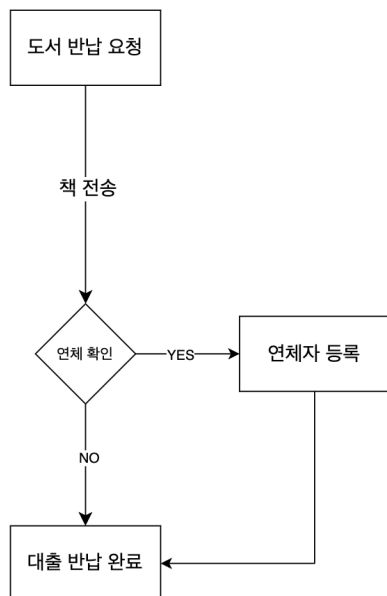
< 시나리오 및 도식도 >

[반납]

[시나리오]

- 반납을 할 때는 책 정보만 가지고 반납할 수 있다.
- 7일 이상의 대출 기간을 주고 이상 초과하면 3일 동안 연체자로 등록되며 이 기간 동안은 대출할 수 없다. (오늘 포함 3일)
- Ex) 2023/ 12/ 02일 연체를 확인했으면 2023/ 12/ 05일부터 다시 책을 빌릴 수 있다.
- 연체자 등록을 할 때는 Redis를 이용하였다. “MEMBER_ID”를 prefix로 이용하였다.

[도식도]

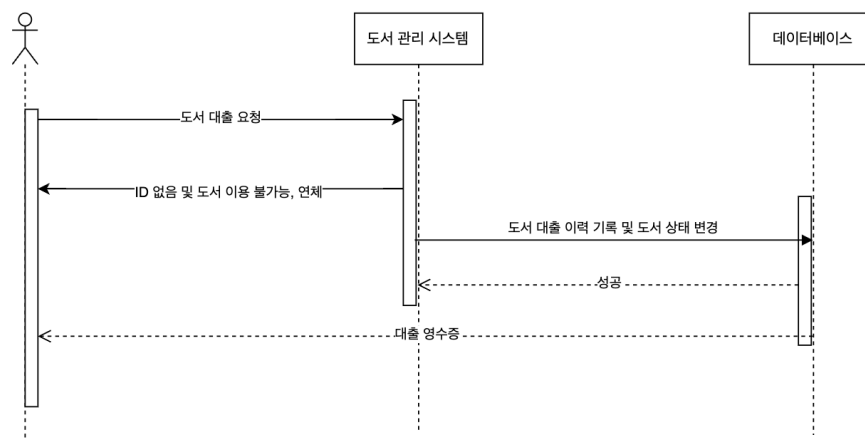
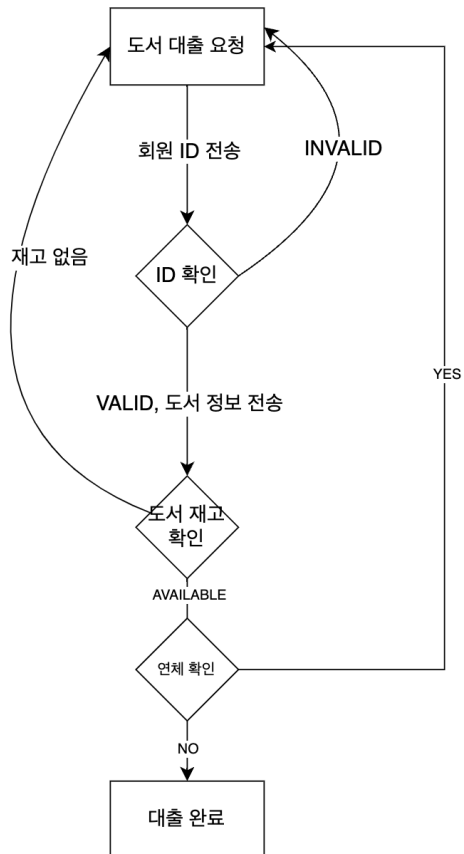


[대출]

[시나리오]

- 대출할 때는 원하는 책과 회원ID를 가지고 대출을 할 수 있다.
- 회원ID가 없으면 도서를 대출 할 수 없다.
- 책이 이용 불가능한 상태이면 대출할 수 없다.
- 연체자라면 연체 기간 동안 책을 빌릴 수 없다.

[도식도]



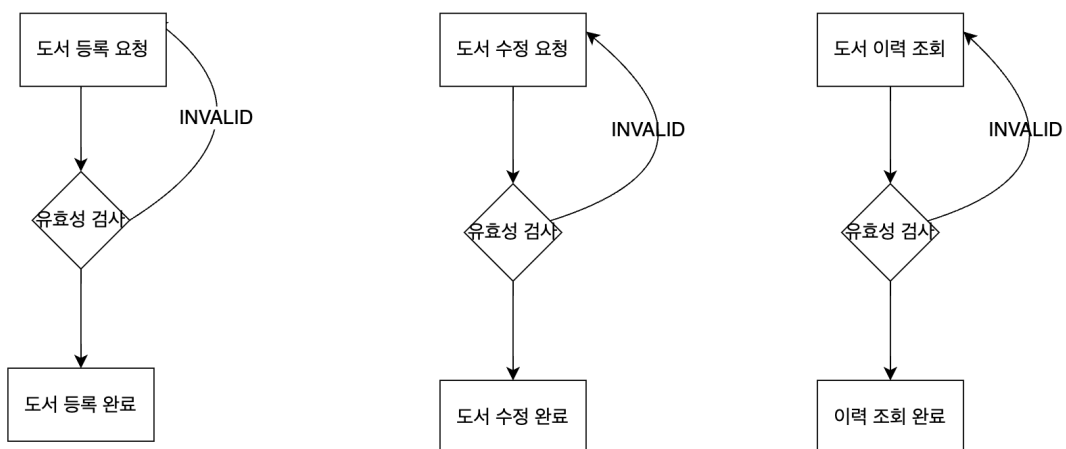
[도서 관리]

[시나리오]

- 책 이름, 작가에 대한 정보를 가지고 도서를 등록한다.
- 책 이름과 작가에 대한 정보를 가지고 도서 정보를 수정할 수 있다.
- 도서 이력 조회 페이지 적용

[도식도]

[도서 관리]



[회원 가입]

[시나리오]

- 방문자는 이메일과 이름으로 회원을 등록한다.
- 방문자는 **UUID**를 회원 아이디로 지급 받는다.

[도식도]

