

REST API

도서 관리 어플리케이션

윤수환

깃허브주소- https://github.com/SuHwanYoon/SpringBootAPI_Library-app.git

AWS배포URL- <http://13.230.93.73:8080/v1/index.html>

목차

1.애플리케이션 개요.....	3p
2.클라이언트 화면구성	4p
3.DB테이블 구성도	5p
• Book테이블 상세	6p
• User테이블 상세	7p
• User_loan_histor테이블 상세	8p
• 테이블 관계	9p
4.애플리케이션 기능과 DB변동	
• 사용자등록.....	10p~11p
• 사용자목록.....	12p
• 사용자수정.....	13p
• 사용자삭제.....	14p~15p
• 책등록.....	16p
• 책대출.....	17p~21p
• 책반납.....	22p
• 반납과 대출 연관동작.....	23p
• 사용자정보와 대출기록의연과관계.....	24p
5. 애플리케이션 동작 문제점	

애플리케이션 개요

1.목표

- SpringBoot기반으로 클라이언트 신호를 받을 RESTAPI도서관리 어플리케이션 EndPoint를구현해 구축 및 배포

2.기능

- RESTAPI로 사용자 등록,삭제,갱신의 CRUD구현
- RESTAPI로 대출,반납의 기반이 될 도서를 등록
- 등록된 사용자의 이름과 도서이름을 기반으로 도서 대출 및 반납

3.개발환경

- WebFrameWork –SpringBoot 2.7.6
- DB –MySQL
- ORM –Spring-Data-JPA

4.배포환경

- AWS EC2

클라이언트 화면구성

사용자 등록

이름

나이

저장

책 등록

책 이름

저장

책 대출

이름

책 이름

저장

책 반납

이름

책 이름

저장

사용자 이름	나이		
윤수환	33세	수정	삭제
김재바	23세	수정	삭제
박민수	45세	수정	삭제
유지영	15세	수정	삭제

4

DB테이블 구성도

1.book 테이블

유저가 도서대출 및 반납을 할때
참조할 도서이름을 등록하는 테이블

2.user 테이블

도서를 대출하고 반납할 유저테이블
이름과 나이정보를 등록한다

3.user_loan_history 테이블

유저가 대출한 도서의 기록 및 상태
가 등록되는 테이블

```
mysql> show tables;
+-----+
| Tables_in_library |
+-----+
| book               |
| user               |
| user_loan_history |
+-----+
3 rows in set (0.00 sec)
```

Book테이블 상세

1. 컬럼 상세 설명

- id- 등록된 도서를 구별할 PK이며 Auto_increment 속성이 설정되어 있어 insert시 자동으로 증가한다

※book테이블과 연결된 Book 엔티티의 id속성은 @GeneratedValue(strategy = GenerationType.IDENTITY) 어노테이션으로 인해 DB에 Book 엔티티 등록시 자동증가

- name- 클라이언트에서 입력받는 도서명

※ Book 엔티티의 name속성은 @Column(nullable = false) 어노테이션으로 인해 null값 입력이 허용되지않음

```
mysql> describe book;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(255)	YES		NULL	

```
2 rows in set (0.00 sec)
```

User테이블 상세

1. 컬럼 상세 설명

- id- 등록된 유저를 구별할 PK이며 Auto_increment 속성이 설정되어 있다

※ User 엔티티의 id 속성은

@GeneratedValue(strategy = GenerationType.IDENTITY)
어노테이션으로 인해 DB에 User 엔티티가
등록될 때 자동증가

- name- 클라이언트에서 입력받아
등록 및 수정할 사용자 이름

※ Book 엔티티의 name 속성은

@Column(nullable = false, length = 20)
어노테이션으로 null 값이 허용되지 않으며
최대 20byte 까지 허용

- age- 클라이언트에서 입력받는
유저의 나이

```
mysql> describe user;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(25)	YES		NULL	
age	int	YES		NULL	

```
3 rows in set (0.00 sec)
```

User_loan_history

테이블 상세

1. 컬럼 상세 설명

- id- 도서대출기록을 구별할 PK이며 Auto_increment 속성이 설정되어 있다

※ UserLoanHistory 엔티티의 id 속성은
`@GeneratedValue(strategy = GenerationType.IDENTITY)`
어노테이션으로 인해 DB에 UserLoanHistory
엔티티 저장시 자동증가

- user_id- 도서를 대출한 유저이름이 들어갈 컬럼

※ User 엔티티의 id 속성을 참조하는 외래키(FK)

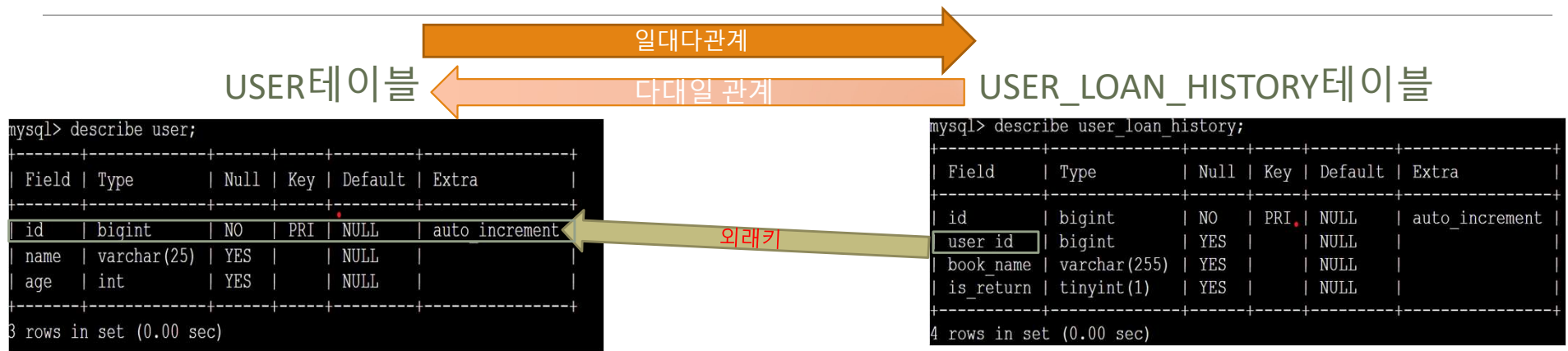
- book_name- 유저가 등록된 도서를 대출시 표시되는 도서이름
- is_return- 도서의 대출상태를 나타내며
대출중인 상태-0(false) 와 반납된 상태-1(true)
값으로 구분된다

```
mysql> describe user_loan_history;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
user_id	bigint	YES	FK	NULL	
book_name	varchar(255)	YES		NULL	
is_return	tinyint(1)	YES		NULL	

4 rows in set (0.00 sec)

테이블 관계



- User 엔티티와 UserLoanHistory 엔티티는 @OneToMany 어노테이션을 사용하여 일대다 관계를 설정하고 있다.
즉, 한 명의 사용자가 여러 권의 도서를 대출할 수 있으며, 각각의 대출 이력은 UserLoanHistory 엔티티로 관리됨.
- UserLoanHistory 엔티티에는 @ManyToOne 어노테이션을 사용하여 User 엔티티와의 다대일 관계를 설정하고 있음.
즉, 각각의 대출 이력은 반드시 한 명의 사용자에게 속해 있어야 함.
- 이러한 관계를 통해, User 엔티티와 UserLoanHistory 엔티티 간의 일대다 관계를 설정하고, 대출 이력을 관리할 수 있음.

애플리케이션 작동과 DB변동

-사용자 등록(POST-http://13.230.93.73:8080/user)

이름,나이입력



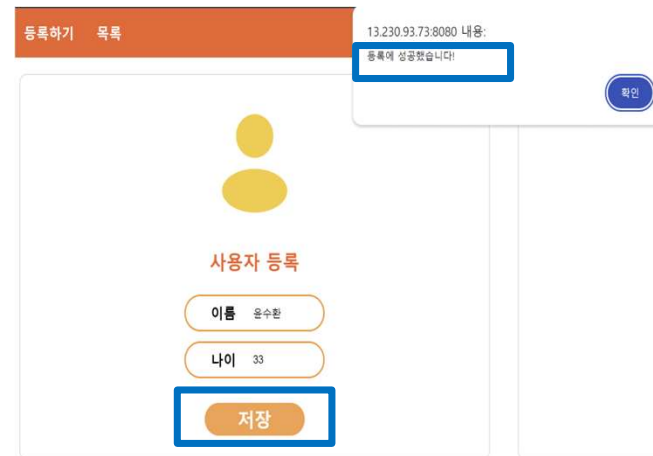
사용자 등록

이름 윤수환

나이 33

저장

저장버튼클릭->등록성공 메세지



등록하기 목록

13.230.93.73:8080 내용:
등록에 성공했습니다!

확인

사용자 등록

이름 윤수환

나이 33

저장

애플리케이션 작동과 DB변동

-사용자 등록성공(POST-http://13.230.93.73:8080/user)

등록성공시 유저 목록에 표시됨

등록하기 목록	
사용자 이름	나이
윤수환	33세
<div>수정 삭제</div>	

DB에는 입력한 사용자이름,나이 그리고 자동증가된 ID컬럼이 1부터 시작하여 INSERT됨

```
mysql> select * from user;
+----+-----+-----+
| id | name   | age  |
+----+-----+-----+
| 1  | 윤수환 | 33   |
+----+-----+-----+
1 row in set (0.00 sec)
```

또다른 사용자 등록시 목록

등록하기 목록	
사용자 이름	나이
윤수환	33세
<div>수정 삭제</div>	
김자바	45세
<div>수정 삭제</div>	

DB에는 입력한 사용자이름,나이 그리고 자동생성된 ID컬럼이 2로 증가되어 INSERT됨

```
mysql> select * from user;
+----+-----+-----+
| id | name   | age  |
+----+-----+-----+
| 1  | 윤수환 | 33   |
| 2  | 김자바 | 45   |
+----+-----+-----+
2 rows in set (0.01 sec)
```

애플리케이션 작동과 DB변동

-사용자 목록 불러오기(GET-http://13.230.93.73:8080/user)

상단 '목록'버튼 클릭시 유저 목록이 표시됨

등록하기 목록			
사용자 이름	나이		
윤수환	33세	수정	삭제
김자바	45세	수정	삭제
홍길동	12세	수정	삭제

USER테이블의 SELECT 문

```
mysql> select * from user;
+----+-----+-----+
| id | name      | age |
+----+-----+-----+
|  1 | 윤수환    |  33 |
|  2 | 김자바    |  45 |
|  3 | 홍길동    |  12 |
+----+-----+-----+
3 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

-사용자 이름수정(PUT-http://13.230.93.73:8080/user)

‘수정’버튼클릭후 수정할 이름을 입력

사용자 정보 수정하기

현재 이름 윤수환

새 이름

윤수환22

수정

수정버튼클릭시 수정된 이름으로 목록에 표시됨

등록하기		목록	
사용자 이름	나이		
윤수환22	33세	수정	삭제
김자바	45세	수정	삭제
홍길동	12세	수정	삭제

변경된 USER테이블의 NAME 컬럼

```
mysql> select * from user;
+----+-----+-----+
| id | name      | age |
+----+-----+-----+
| 1  | 윤수환22  | 33  |
| 2  | 김자바    | 45  |
| 3  | 홍길동    | 12  |
+----+-----+-----+
3 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

-사용자 정보삭제(DELETE-http://13.230.93.73:8080/user)

‘삭제’버튼클릭시사용자 이름이 삭제됨

등록하기 목록	
사용자 이름	나이
윤수환22	33세
김자바	45세
홍길동	12세



등록하기 목록	
사용자 이름	나이
윤수환22	33세
김자바	45세

USER테이블 에서 삭제됨

```
mysql> select * from user;
+----+-----+-----+
| id | name      | age |
+----+-----+-----+
| 1  | 윤수환 22 | 33  |
| 2  | 김자바    | 45  |
| 3  | 홍길동    | 12  |
+----+-----+-----+
3 rows in set (0.00 sec)
```



```
mysql> select * from user;
+----+-----+-----+
| id | name      | age |
+----+-----+-----+
| 1  | 윤수환 22 | 33  |
| 2  | 김자바    | 45  |
+----+-----+-----+
2 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

-사용자 정보삭제(DELETE-http://13.230.93.73:8080/user)

‘홍길동’을 삭제하고 ‘김철수’를 추가

홍길동의 PK ID ‘3’은 사라지고 김철수가 4번째로 추가됐음을 알수있다

등록하기 목록

사용자 이름 나이

윤수환 33세

수정

삭제

김자바 45세

수정

삭제

김철수 12세

수정

삭제

```
mysql> select * from user;
+----+-----+-----+
| id | name      | age  |
+----+-----+-----+
| 1  | 윤수환    | 33   |
| 2  | 김자바    | 45   |
| 4  | 김철수    | 12   |
+----+-----+-----+
3 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

-책이름 등록성공&실패 케이스

(POST-http://13.230.93.73:8080/book)

책이름 입력후 등록성공

A screenshot of a web application interface for book registration. At the top, a status bar shows the IP '13.230.93.73:8080' and the message '책 등록에 성공했습니다!' (Book registration successful!). Below this is a blue '확인' (Check) button. The main content area features a yellow book icon, the text '책 등록' (Book Registration), a text input field containing '책 이름' (Book Name) with a placeholder '스프링의정석' (Spring's Standard), and an orange '저장' (Save) button.

DB에는 책이름 그리고 자동생성된 ID컬럼이 1부터 시작하여 INSERT됨

```
mysql> select * from book;
+----+-----+
| id | name  |
+----+-----+
| 1  | 스프링의정석 |
+----+-----+
1 row in set (0.00 sec)
```

책이름을 입력하지 않으면 등록실패

A screenshot of the same book registration form, but with a failure message. The status bar shows the IP '13.230.93.73:8080' and the message '서버 내부 오류입니다.' (Server internal error). Below this is a blue '확인' (Check) button. The main content area features a yellow book icon, the text '책 등록' (Book Registration), a text input field that is empty (highlighted with a red box), and an orange '저장' (Save) button.

애플리케이션 작동과 DB변동

-책대출성공케이스

(POST-http://13.230.93.73:8080/book/loan)

사용자 목록에 존재하는 이름 & 등록되어있는 책이름으로 대출 신청

10/v1/index.html

13.230.93.73:8080 내용:
책 대출에 성공했습니다!

확인

사용자 등록

이름

나이

저장

책 대출

이름 윤수환

책 이름 스프링의정석

저장

사용자는 여러 개의 책을 대출할수있으며

대출기록은 USER_LOAN_HISTORY테이블에 담긴다

```
mysql> select * from user_loan_history;
```

id	user_id	book_name	is_return
1	1	스프링의정석	0
2	1	파이썬의정석	0
3	1	자바의정석	0
4	2	혼공자바	0

4 rows in set (0.00 sec)

애플리케이션 작동과 DB변동

-책대출성공케이스

(POST-http://13.230.93.73:8080/book/loan)

USER테이블에 존재하는 이름에 해당하는
ID값으로 대출기록이 남는다

```
mysql> select * from user;
```

id	name	age
1	윤수환	33
2	김자바	45
4	김철수	12

3 rows in set (0.00 sec)

USER_ID가 '1'인 윤수환은 3권을, USER_ID가 '2'인 김자바는 1권을 대출했으며
IS_RETURN컬럼 0(FALSE값)은 현재 해당 도서가 대출 상태임을 의미한다

```
mysql> select * from user_loan_history;
```

id	user_id	book_name	is_return
1	1	스프링의정석	0
2	1	파이썬의정석	0
3	1	자바의정석	0
4	2	혼공자바	0

4 rows in set (0.00 sec)

애플리케이션 작동과 DB변동

-책대출실패케이스

(POST-http://13.230.93.73:8080/book/loan)

사용자 목록에 **존재하는 않는 이름** & **등록되어있는 책이름**으로 대출 신청

등록하기 목록

13.230.93.73:8080 내용: 서버 내부 오류입니다. 확인

사용자 등록

이름

나이

저장

책 대출

이름 김무명

책 이름 스프링의정석

저장

BOOK테이블에 '스프링의정석'도서는 존재하지만

USER테이블에 김무명이라는 사용자는 없기 때문에 대출이 불가

```
mysql> select * from book;
```

id	name
1	스프링의정석
2	파이썬의정석
3	자바의정석
4	혼공자바
5	클린코드
6	네트워크기본

6 rows in set (0.00 sec)

```
mysql> select * from user;
```

id	name	age
1	윤수환	33
2	김자바	45
4	김철수	12

3 rows in set (0.00 sec)

애플리케이션 작동과 DB변동

-책대출실패케이스

(POST-http://13.230.93.73:8080/book/loan)

사용자 목록에 존재하는 이름 & 등록되어있지 않는 책이름으로 대출 신청

사용자 등록

이름

나이

저장

책 대출

이름

책 이름

저장

USER테이블에 '윤수환' 사용자는 존재하지만

BOOK테이블에 '수학의정석' 도서는 존재하지않기때문에 대출이 불가

```
mysql> select * from user;
+----+-----+-----+
| id | name  | age  |
+----+-----+-----+
| 1  | 윤수환 | 33   |
| 2  | 김사바 | 45   |
| 4  | 김철수 | 12   |
+----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from book;
+----+-----+
| id | name                |
+----+-----+
| 1  | 스프링의정석        |
| 2  | 파이썬의정석        |
| 3  | 자바의정석          |
| 4  | 흔공자바            |
| 5  | 클린코드            |
| 6  | 네트워킹기본        |
+----+-----+
6 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

-책대출실패케이스

(POST-http://13.230.93.73:8080/book/loan)

존재하는 사용자가 이미 다른사용자가 대출한책을 대출 신청 하는 케이스

‘김자바’사용자는 USER테이블에 존재하지만, 스프링의 정석도서는

이미 ‘윤수환’ 사용자가 대출한 상태 0 (IS_RETURN==FALSE)이므로 대출불가

```
mysql> select * from user;
```

id	name	age
1	윤수환	33
2	김자바	45
4	김철수	12

3 rows in set (0.00 sec)

```
mysql> select * from user_loan_history;
```

id	user_id	book_name	is_return
1	1	스프링의정석	0
2	1	파이썬의정석	0
3	1	자바의정석	0
4	2	혼공자바	0

4 rows in set (0.00 sec)

애플리케이션 작동과 DB변동

-책반납성공케이스

(POST-http://13.230.93.73:8080/book/return)

사용자 이름 & 해당 사용자가 대출한 책이름으로 반납 신청

책 반납이 성공했습니다!

확인

책 등록

책 이름

저장

책 반납

이름 윤수환

책 이름 스프링의정석

저장

USER_ID'1'인 '윤수환' 사용자가 '스프링의정석' 도서를 반납하고

ID_RETURN 컬럼값 1(TRUE값)은 반납완료 상태를 의미한다

```
mysql> select * from user;
+----+-----+-----+
| id | name   | age  |
+----+-----+-----+
| 1  | 윤수환 | 33   |
| 2  | 김사마 | 45   |
| 4  | 김철수 | 12   |
+----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from user_loan_history;
+----+-----+-----+-----+
| id | user_id | book_name | is_return |
+----+-----+-----+-----+
| 1  | 1       | 스프링의정석 | 1         |
| 2  | 1       | 파이썬의정석 | 0         |
| 3  | 1       | 자바의정석   | 0         |
| 4  | 2       | 혼공자바     | 0         |
| 5  | 2       | 클린코드     | 0         |
+----+-----+-----+-----+
```

애플리케이션 작동과 DB변동

- 반납과 대출의 연관동작

(POST-http://13.230.93.73:8080/book/loan)

사용자가 반납한 책을 또다른 사용자가 대출

ChatGPT 일반채용 < 모집공... 티비위키 13.230.93.73:8080 내용: 책 대출에 성공했습니다!

확인

사용자 등록

이름

나이

저장

책 대출

이름 김자바

책 이름 스프링의정석

저장

USER_ID'1'인'윤수환'사용자가 반납해 ID_RETURN 컬럼값 1(TRUE값) 인 '스프링의정석'도서를

USER_ID'2'인'김자바'사용자가 대출하고 스프링의정석도서는 다시 대출상태가 된다

```
mysql> select * from user;
+----+-----+-----+
| id | name   | age  |
+----+-----+-----+
| 1  | 유수환 | 33   |
| 2  | 김자바 | 45   |
| 4  | 김철수 | 12   |
+----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from user_loan_history;
+----+-----+-----+-----+
| id | user_id | book_name | is_return |
+----+-----+-----+-----+
| 1  | 1       | 스프링의정석 | 1         |
| 2  | 1       | 파이썬의정석 | 0         |
| 3  | 1       | 자바의정석   | 0         |
| 4  | 2       | 혼공자바     | 0         |
| 5  | 2       | 크리쿠드     | 0         |
| 6  | 2       | 스프링의정석 | 0         |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

애플리케이션 작동과 DB변동

- 사용자정보와 대출기록의 연관관계 (DELETE-http://13.230.93.73:8080/user)

네트워크 기본 도서를 대출한 USER_ID가 '4'인 '김철수' 사용자를 삭제했을 경우

사용자 이름	나이	수정	삭제
윤수환	33세		
김자바	45세		
김철수	12세		
김영희	44세		

```
mysql> select * from user;
+----+-----+-----+
id | name   | age  |
+----+-----+-----+
1 | 윤수환 | 33   |
2 | 김자바 | 45   |
4 | 김철수 | 12   |
6 | 김영희 | 44   |
+----+-----+-----+
rows in set (0.00 sec)
```

사용자 정보삭제시

id	name	age
1	윤수환	33
2	김자바	45
6	김영희	44

id	user_id	book_name	is_return
1	1	스프링의정석	1
2	1	파이썬의정석	0
3	1	자바의정석	0
4	2	혼공자바	0
5	2	클린코드	0
6	2	스프링의정석	0
7	4	네트워크기본	0
8	6	리액트의정석	0

대출기록도 사라짐

id	user_id	book_name	is_return
1	1	스프링의정석	1
2	1	파이썬의정석	0
3	1	자바의정석	0
4	2	혼공자바	0
5	2	클린코드	0
6	2	스프링의정석	0
8	6	리액트의정석	0

USER엔티티와 USER_LOAN_HISTORY엔티티는

[ORPHANREMOVAL = TRUE]속성으로 관계되어있기 때문에

김철수 사용자 삭제시 해당 사용자가대출한기록도 자동삭제됨