

Project

API 설계

회원 API 설계

1. 회원가입

Method	POST
URI	/join
HTTP status code	성공 200
Request Body	{ email : "사용자가 입력한 이메일" password : "사용자가 입력한 비밀번호" }
Response Body	

2. 로그인

Method	POST
URI	/login
HTTP status code	성공 200
Request Body	{ email : "사용자가 입력한 이메일" password : "사용자가 입력한 비밀번호" }
Response Body	JWT Token

3. 비밀번호 초기화 요청

Method	POST
URI	/reset
HTTP status code	성공 200

Request Body	{ email : "사용자가 입력한 이메일" }
Response Body	

4. 비밀번호 초기화(=수정)

Method	PUT
URI	/reset
HTTP status code	성공 200
Request Body	{ password : "사용자가 입력한 비밀번호" }
Response Body	

도서 API 설계

1. 상품 전체 조회

//추가적으로 경로 , 8개씩 보내줘야함

Method	GET
URI	/books
HTTP status code	성공 200
Request Body	
Response Body	{ { title : "도서제목", summary : "요약 설명 ", author : "도서 작가 " price : 가격 likes : 좋아요 수 pubDate : "출간일" } { title : "도서제목", summary : "요약 설명 ",

	<pre>author : "도서 작가 " price : 가격 likes : 좋아요 수 pubDate : "출간일" } }</pre>
--	---

2. 개별 도서 조회

//추가적으로 경로

Method	GET
URI	/books/{bookId}
HTTP status code	성공 200
Request Body	
Response Body	<pre>{ { id : 도서id, title : "도서제목", category : "카테고리", format : "포맷", isbn : "isbn", summary : "요약 설명 ", description : "상세 설명", author : "도서 작가 ", pages : "쪽 수" index : "목차", price : 가격 likes : 좋아요 수 liked : boolean pubDate : "출간일" } }</pre>

3. 카테고리별 도서 목록 조회

//추가적으로 이미지 경로 ,8개씩
//카테고리 아이디는 어떻게 알고 보내줄까?

Method	GET
URI	/books ? categoryId={categoryId}&new={boolean}

HTTP status code	성공 200
Request Body	
Response Body	<pre>{ { title : "도서제목", summary : "요약 설명 ", author : "도서 작가 " price : 가격 likes : 좋아요 수 pubDate : "출간일" } { title : "도서제목", summary : "요약 설명 ", author : "도서 작가 " price : 가격 likes : 좋아요 수 pubDate : "출간일" } }</pre>

좋아요 API 설계

1. 좋아요 추가 ERD 그려본 후 다시 설계예정

Method	Post
URI	/likes/{bookId}
HTTP status code	성공 200
Request Body	<pre>{ id }</pre>
Response Body	

장바구니 API 설계

1. 장바구니 담기

Method	POST
URI	/cart
HTTP status code	성공 201
Request Body	{ bookId : 도서 id, count : 수량 }
Response Body	

2. 장바구니 조회

Method	GET
URI	/cart
HTTP status code	성공 200
Request Body	
Response Body	[{ cartitemId : 장바구니 도서 id bookId : 도서 id, title : "도서 제목", summary : "도서 요약", price : "가격", count : 수량 } { cartitemId : 장바구니 도서 id bookId : 도서 id, title : "도서 제목", summary : "도서 요약", price : "가격" count : 수량 } ...]

3. 장바구니 도서 삭제

Method	DELETE
--------	--------

URI	/cart/{bookId}
HTTP status code	성공 200
Request Body	
Response Body	<pre>[{ bookId : 도서 id, title : "도서 제목", summary : "도서 요약", price : "가격", count : 수량 } { bookId : 도서 id, title : "도서 제목", summary : "도서 요약", price : "가격" count : 수량 } ...]</pre>

4. 장바구니에서 선택한 상품 목록 조회

Method	GET
URI	/
HTTP status code	성공 200
Request Body	[cartItemId, cartItemId,...]
Response Body	<pre>[{ cartItemId : 장바구니 도서 id bookId : 도서 id, title : "도서 제목", summary : "도서 요약", price : "가격", count : 수량 } { cartItemId : 장바구니 도서 id bookId : 도서 id,</pre>

```

title : "도서 제목",
summary : "도서 요약",
price : "가격"
count : 수량
}
...
]

```

주문 API 설계

1. 결제하기 = 주문하기

주문된 상품은 삭제되어야 함.

Method	POST
URI	/orders
HTTP status code	성공 200
Request Body	<pre> { items : [{ cartItemId : 장바구니 도서 id bookId : 도서 id count : 수량 } { cartItemId : 장바구니 도서 id bookId : 도서 id count : 수량 }] delivery : { address : "주소" receiver : "이름 " contact "010-0000-0000" } totalPrice : 총 금액 } </pre>
Response Body	<pre> [</pre>

2. 주문목록 조회

Method	GET
URI	/orders
HTTP status code	성공 200
Request Body	
Response Body	<pre>[{ created_at : "주문일자" delivery : { address : "주소" receiver : "이름" contact : "전화번호" } bookTitle : "대표 책 제목", titlePrice : 결제금액 totalCount : 총수량 }]</pre>

3. 주문 상세 조회

Method	GET
URI	/orders/{orderId}
HTTP status code	성공 200
Request Body	
Response Body	<pre>[{ bookId : 도서 id, bookTitle : "도서 제목" author : "작가명", price : 가격 count : 수량 }]</pre>

