

Shop API

Version 0.0.1-SNAPSHOT

목차

1. Brand

- 1.1. 브랜드 조회
- 1.2. 브랜드 생성

2. Product

- 2.1. 상품 목록 조회
- 2.2. 상품 조회
- 2.3. 상품 생성
- 2.4. 상품 수정
- 2.5. 상품 삭제

3. 검색

- 3.1. 카테고리 별 최저가격 브랜드와 상품 가격, 총액을 조회하는 API
- 3.2. 단일 브랜드로 모든 카테고리 상품을 구매할 때 최저가격에 판매하는 브랜드와 카테고리의 상품가격, 총액을 조회하는 API
- 3.3. 카테고리 이름으로 최저, 최고 가격 브랜드와 상품 가격을 조회하는 API

1. Brand

1.1. 브랜드 조회

curl-request

```
$ curl 'http://localhost:8080/v1/brands/1' -i -X GET
```

BASH

http-request

```
GET /v1/brands/1 HTTP/1.1
Host: localhost:8080
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 28

{"id":1,"name":"Test Brand"}
```

HTTP

Table 1. /v1/brands/{id}

Parameter	Description
id	ID

Table 2. response-fields

Path	Type	Description
id	Number	ID
name	String	브랜드 이름

1.2. 브랜드 생성

curl-request

```
$ curl 'http://localhost:8080/v1/brands' -i -X POST \
-H 'Content-Type: application/json;charset=UTF-8' \
-d '{"name":"test"}
```

BASH

http-request

```
POST /v1/brands HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 15
Host: localhost:8080

{"name":"test"}
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 28

{"id":1,"name":"Test Brand"}
```

Table 3. request-fields

Path	Type	Description
name	String	브랜드 이름

Table 4. response-fields

Path	Type	Description
id	Number	ID
name	String	브랜드 이름

2. Product

2.1. 상품 목록 조회

curl-request

\$ curl 'http://localhost:8080/v1/products?brandName=A' -i -X GET

BASH

http-request

GET /v1/products?brandName=A HTTP/1.1
Host: localhost:8080

HTTP

http-response

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 84

HTTP

[{"id":1,"brandId":1,"brandName":"Test Brand","category":"가방","price":"10,000"}]

Table 5. query-parameters

Parameter	Description
brandName	브랜드 이름

Table 6. response-fields

Path	Type	Description
[].id	Number	ID
[].brandId	Number	브랜드 ID
[].brandName	String	브랜드 이름
[].category	String	카테고리
[].price	String	상품 가격

2.2. 상품 조회

curl-request

\$ curl 'http://localhost:8080/v1/products/1' -i -X GET

BASH

http-request

GET /v1/products/1 HTTP/1.1
Host: localhost:8080

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 82
```

HTTP

```
{"id":1,"brandId":1,"brandName":"Test Brand","category":"가방","price":"10,000"}
```

Table 7. /v1/products/{id}

Parameter	Description
id	ID

Table 8. response-fields

Path	Type	Description
id	Number	ID
brandId	Number	브랜드 ID
brandName	String	브랜드 이름
category	String	카테고리
price	String	상품 가격

2.3. 상품 생성

curl-request

```
$ curl 'http://localhost:8080/v1/products' -i -X POST \
-H 'Content-Type: application/json;charset=UTF-8' \
-d '{"brandId":1,"category":"가방","price":10000}'
```

BASH

http-request

```
POST /v1/products HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 47
Host: localhost:8080

{"brandId":1,"category":"가방","price":10000}
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 82

{"id":1,"brandId":1,"brandName":"Test Brand","category":"가방","price":"10,000"}
```

HTTP

Table 9. request-fields

Path	Type	Description
brandId	Number	브랜드 ID
category	String	카테고리

Path	Type	Description
price	Number	상품 가격

Table 10. response-fields

Path	Type	Description
id	Number	ID
brandId	Number	브랜드 ID
brandName	String	브랜드 이름
category	String	카테고리
price	String	상품 가격

2.4. 상품 수정

curl-request

```
$ curl 'http://localhost:8080/v1/products/1' -i -X PUT \  
-H 'Content-Type: application/json;charset=UTF-8' \  
-d '{"category":"가방","price":20000}'
```

BASH

http-request

```
PUT /v1/products/1 HTTP/1.1  
Content-Type: application/json;charset=UTF-8  
Content-Length: 35  
Host: localhost:8080  
  
{"category":"가방","price":20000}
```

HTTP

http-response

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: 82  
  
{"id":1,"brandId":1,"brandName":"Test Brand","category":"가방","price":10,000}
```

HTTP

Table 11. /v1/products/{id}

Parameter	Description
id	ID

Table 12. request-fields

Path	Type	Description
category	String	카테고리
price	Number	상품 가격

Table 13. response-fields

Path	Type	Description
id	Number	ID
brandId	Number	브랜드 ID
brandName	String	브랜드 이름
category	String	카테고리
price	String	상품 가격

2.5. 상품 삭제

curl-request

```
$ curl 'http://localhost:8080/v1/products/1' -i -X DELETE
```

BASH

http-request

```
DELETE /v1/products/1 HTTP/1.1
Host: localhost:8080
```

HTTP

http-response

```
HTTP/1.1 204 No Content
```

HTTP

Table 14. /v1/products/{id}

Parameter	Description
id	ID

3. 검색

3.1. 카테고리 별 최저가격 브랜드와 상품 가격, 총액을 조회하는 API

curl-request

```
$ curl 'http://localhost:8080/v1/search/min-product-by-category' -i -X GET
```

BASH

http-request

```
GET /v1/search/min-product-by-category HTTP/1.1
Host: localhost:8080
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 91

{"list":[{"category":"가방","brandName":"Test Brand","price":"10,000"}],"total":"10,000"}
```

HTTP

Table 15. response-fields

Path	Type	Description
list	Array	상품 리스트
list[].category	String	카테고리
list[].brandName	String	브랜드명
list[].price	String	상품 가격
total	String	총 가격

3.2. 단일 브랜드로 모든 카테고리 상품을 구매할 때 최저가격에 판매하는 브랜드와 카테고리의 상품가격, 총액을 조회하는 API

curl-request

```
$ curl 'http://localhost:8080/v1/search/min-brand' -i -X GET
```

BASH

http-request

```
GET /v1/search/min-brand HTTP/1.1
Host: localhost:8080
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 102

{"min":{"brandName":"Test Brand","categoryList":[{"category":"가방","price":"10,000"}],"total":"0"}}
```

HTTP

Table 16. response-fields

Path	Type	Description
min.brandName	String	브랜드 이름
min.categoryList	Array	상품 리스트
min.categoryList[].category	String	카테고리
min.categoryList[].price	String	상품 가격
min.total	String	총 가격

3.3. 카테고리 이름으로 최저, 최고 가격 브랜드와 상품 가격을 조회하는 API

curl-request

```
$ curl 'http://localhost:8080/v1/search/min-max-price' -i -X POST \
-H 'Content-Type: application/json;charset=UTF-8' \
-d '{"category":"상의"}'
```

BASH

http-request

```
POST /v1/search/min-max-price HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 21
Host: localhost:8080

{"category":"상의"}
```

HTTP

http-response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 121

{"category":"상의","min":{"brandName":"Test Brand","price":"10,000"},"max":{"brandName":"Test Brand","price":"10,C
```

HTTP

Table 17. request-fields

Path	Type	Description
category	String	카테고리

Table 18. response-fields

Path	Type	Description
category	String	카테고리
min.brandName	String	최저가 브랜드명
min.price	String	최저가 상품 가격
max.brandName	String	최고가 브랜드명
max.price	String	최고가 상품 가격

Version 0.0.1-SNAPSHOT

Last updated 2024-08-28 11:40:42 +0900