

Simple eCommerce Application

1. 개요

Product와 Category를 관리하는 Rest API를 구현한다.

2. 가정

- 1) Product는 다수개의 Category에 속할 수 있다. 즉, 전동 칫솔은 "Electronics" 와 "Beauty & Personal Care" 카테고리로 분류될 수 있다.
- 2) Category는 다수개의 SubCategory를 가질 수 있다. 예를 들어 "Electronics" category는 'Audio & Video Components', 'Camera & Photo', 'Computers' SubCategory를 가질 수 있다.

3. Rest API

Products:

- 1) Product CRUD (URL: /api/products), ProductController.java

URL	설명
GET, /api/products	모든 product를 조회한다
GET, /api/products/{id}	특정 id를 가진 product를 조회한다
POST, /api/products Body { "name": "P1", "price": 100.00 }	하나의 product을 생성한다
PUT, /api/products/{id} Body { "name": "P1", "price": 100.00 }	하나의 product을 수정한다.
DELETE, /api/products/{id}	특정 id를 가진 product를 삭제한다

Categories:

1) category CRUD (URL: /api/categories), `CategoryController.java`

URL	설명
GET, /api/categories	모든 category를 조회한다
GET, /api/categories/{id}	특정 id를 가진 category를 조회한다
POST, /api/categories Body { "name": "C1" }	하나의 category를 생성한다
PUT, /api/categories/{id} Body { "name": "C1" }	하나의 category를 수정한다.
DELETE, /api/categories/{id}	특정 id를 가진 category를 삭제한다

2) Add / Remove child categories `CategorySubcategoriesController.java`

URL	설명
GET, /api/categories/{parentid}/subcategories	특정 parentid를 가진 카테고리에 속한 자식카테고리를 조회한다
POST, /api/categories/{parentid}/subcategories/{childid}	Parent category와 child category를 연결한다
DELETE, /api/categories/{parentid}/subcategories/{childid}	Parent category에서 child category를 제거한다

3) Link / Unlink products, `CategoryProductsController.java`

URL	설명
GET, /api/categories/{categoryid}/products	특정 id를 가진 category에 속한 모든 product를 조회한다
POST, /api/categories/{categoryid}/products/{productid}	Product를 category에 넣는다
DELETE, /api/categories/{categoryid}/products/{productid}	Product를 category에서 제거한다

4. 데이터 베이스 초기화

- 1) 데이터베이스를 생성한다: 이름은 ecommerce
- 2) 데이터베이스에 초기 데이터를 세팅한다.

data.sql 이 있는 폴더에서 아래 실행한 후, Mysql Workbench에서 확인

```
$ mysql -u root -p
```

```
$ use ecommerce;
```

```
$ source data.sql;
```

5. 수정해야 할 파일 리스트

기본 소스는 github에 있으며 아래 소스를 수정하기 바람

(<https://github.com/stelladream/eCommerce>)

- 1) 설정 파일: dao-context.xml, service-context.xml, servlet-context.xml
- 2) 컨트롤러 파일: ProductController.java, CategoryControler.java,
CategorySubcategoriesController.java, CategoryProductsController.java

6. 제출 내용

- 1) 소스 파일
- 2) 8번 Rest API에 있는 모든 URL을 실행시켰을 때 postman 스크린샷
위 내용을 담은 보고서(pdf).

7. 종료시한

5월 28일(일) 오후 11시 59분까지

8. Rest API

Products:

1) Get, `http://localhost:8080/ecommerce/api/products`

2) Get, `http://localhost:8080/ecommerce/api/products/1`

3) Post, `http://localhost:8080/ecommerce/api/products`, id 37일 경우

```
{  
    "name": "삼성컴퓨터",  
    "price": 100  
} // body는 postman에서 raw-json format으로 전송
```

4) Put, `http://localhost:8080/ecommerce/api/products/37`

```
{  
    "name": "LG컴퓨터",  
    "price": 100  
}
```

5) Delete, `http://localhost:8080/ecommerce/api/products/37`

Category:

1-1) Get, <http://localhost:8080/ecommerce/api/categories>

1-2) Get, <http://localhost:8080/ecommerce/api/categories/1>

1-3) Post, <http://localhost:8080/ecommerce/api/categories>, id 18 카테고리일 경우

```
{  
  "name": "스마트폰"  
}
```

1-4) Put, <http://localhost:8080/ecommerce/api/categories/18>

```
{  
  "name": "스마트폰"  
}
```

1-5) Delete, <http://localhost:8080/ecommerce/api/categories/18>

2-1) Get, <http://localhost:8080/ecommerce/api/categories/1/subcategories>

2-2) subcategory를 생성한 후(id=19), category(id=1)에 연결한다

Post, <http://localhost:8080/ecommerce/api/categories>

```
{  
  "name": "스마트폰"  
}
```

Post, <http://localhost:8080/ecommerce/api/categories/1/subcategories/19>

2-3) Delete, <http://localhost:8080/ecommerce/api/categories/1/subcategories/19>

3-1) Get, <http://localhost:8080/ecommerce/api/categories/8/products>

먼저 “Computer” 카테고리에 존재하는 Product조회한다

3-2) Product(id=38)를 생성한 후, “Computer” 카테고리에 저장한다.

Post, , <http://localhost:8080/ecommerce/api/products>

```
{  
    "name": "LG컴퓨터",  
    "price": 100  
}
```

Post, <http://localhost:8080/ecommerce/api/categories/8/products/38>

3-3) Delete, <http://localhost:8080/ecommerce/api/categories/8/products/38>