**신한 프로디지털아카데미 Spring 프로젝트**

**주제: 쇼핑몰**

(모든 기능 나열 -> 특정 부피로 분류)

**Response Format**

1. **성공**

**{**

**“success” : True,**

**“response” : 응답 데이터(객체),**

**“error” : null**

**}**

1. **실패**

**{**

**“success” : False,**

**“response” : null,**

**“error” : {**

**“message”: “에러 메시지”,**

**“status”: HTTP status**

**}**

**}**

**Domain Driven Design(DDD)**

* 상품
  + **Create : 상품 등록**
    - 상품 개별 등록 -> URL에 id X
  + **Read : 상품 조회**
    - 상품 전체 조회 -> 4개(paging)
    - **// 상품 카테고리별 조회 -> 카테고리, 4개**  
      카테고리는 idx – 카테고리명

|  |  |
| --- | --- |
| id | category\_name |
| 0 | cup |
| 1 | Tshirt |
| 2 | Pants |

* + - 상품 개별 조회 -> 자세히

**// 1) 전체 조회 & 카테고리별 조회 API 통합? Or 분리?**

// 2) 전체 조회 & 개별 조회 : 상품 데이터 컬럼 차이?

* + **Update : 상품 수정 (PUT)**
    - 상품 개별 수정 -> 데이터 수정
    - // 상품 여러 개 수정 -> 판매 상태 (품절, 판매중지..)
  + **Delete : 상품 삭제**
    - 상품 개별 삭제
    - 상품 여러 개 삭제  
      - id 여러 개 어떻게 담아 보내지?
* 주문
  + CRUD
* 장바구니
  + CRUD
* 회원
  + 회원가입
  + 로그인
  + 회원정보 수정
  + 회원 탈퇴 cf. soft delete

1. **상품 도메인**
2. **상품 개별 등록**

|  |  |
| --- | --- |
| **Method** | POST |
| **URL** | /products |
| **HTTP status code** | 성공 201(created)  실패 400 “**유효성 검사** 결과”  **401 403** : 인증 인가  404 : URL로 갖다 줄 수 있는 게 없을 때  413 415 |
| **Request Body** | // JSON  {  “name” : “상품명”, // ex. “삼성노트북 15” -> swagger API  “price” : 가격,  “description” : “상품 설명” // “summary” : “상품 설명 요약”,  “categoryId” : 카테고리id  } |
| **Response Body** | 400일 때  1.  {   1. “message” : (field) must be (type, sample). Input (field) is \_\_\_\_” 2. “message” : (field) is empty. (field) must be provi…   }  2.  {  “message” : \_\_\_\_\_\_\_,  “errors” : [  {  “field” : “name”,  “received” : \_\_\_\_\_,  “expected” : \_\_\_\_\_  }, {  …  }  ]  } |

1. **상품 전체/카테고리별 조회**

* **전체 조회 : URL / categoryId= null**
* **카테고리별 조회 : URL / categoryId = {categoryId}**

|  |  |
| --- | --- |
| **Method** | GET |
| **URL** | /products?limit={page당 상품수}&currentPage={현재 page}**&categoryId={categoryId}** |
| **HTTP status code** | 성공 200 / 실패 404 |
| **Request Body** |  |
| **Response Body** | // 전체 상품 목록에서는 상품의 상세 정보를 모두 포함  {  products : [  {  “id” : 상품 id,  “name” : “상품명”,  “price” : 가격,  “description” : “상품 설명”,  “categoryId” : 카테고리id  }, {  “id” : 상품 id,  “name” : “상품명”,  “price” : 가격,  “description” : “상품 설명”,  “categoryId” : 카테고리id  }, …  ],  pagination : {  “currentPage” : 현재 페이지,  “totalCount” : 총 상품 수  }  } |

1. **상품 개별 조회**

|  |  |
| --- | --- |
| **Method** | GET |
| **URL** | /products/{id} |
| **HTTP status code** | 성공 200 / 실패 404 |
| **Request Body** |  |
| **Response Body** | {  “id” : 상품 id,  “name” : “상품명”,  “price” : 가격,  “description” : “상품 설명”,  “categoryId” : 카테고리id  } |

1. **상품 개별 수정**

|  |  |
| --- | --- |
| **Method** | PUT |
| **URL** | /products/{id} |
| **HTTP status code** | 성공 200 / 실패 400 |
| **Request Body** | {  “id” : 상품 id,  “name” : “상품명”,  “price” : 가격,  “description” : “상품 설명”,  “categoryId” : 카테고리id  } |
| **Response Body** |  |

1. **상품 한 개 삭제**

|  |  |
| --- | --- |
| **Method** | DELETE |
| **URL** | /products/{id} |
| **HTTP status code** | 성공 200 / 실패 400 |
| **Request Body** | - |
| **Response Body** | - |

1. **상품 여러 개 삭제**

* **DELETE + body [배열] : 웹 브라우저 따라 무시하는 애들도..**
* **URL parameter : 안정성 (GET 제외)**

|  |  |
| --- | --- |
| **Method** | POST (body로 값을 숨기고 싶을 때) |
| **URL** | /products/delete  /delete-products |
| **HTTP status code** | 성공 200 / 실패 400 |
| **Request Body** | {  “productIds”: [1,2,3]  } |
| **Response Body** |  |

1. **회원 도메인**
2. **회원가입**

|  |  |
| --- | --- |
| **Method** | POST |
| **URL** | /signup or /join |
| **HTTP status code** | 성공 : 201  에러   * 중복된 아이디: 409 |
| **Request Body** | {  “user\_id” : “사용자 아이디”,  “pw”: “비밀번호”, cf. 시큐어 코딩  “name”: “이름”,  “email”: “이메일”,  “contact”: “연락처”  } |
| **Response Body** | {  “user\_id”: “사용자 아이디”  } |

1. **로그인**

|  |  |
| --- | --- |
| **Method** | POST |
| **URL** | /login |
| **HTTP status code** | 200 / 400 |
| **Request Body** | {  “user\_id”: “사용자 아이디”,  “profile\_info”: “비밀번호”  } |
| **Response Body** | JWT 문자열 cf. jwt.io(+ 인증: 전체 출입문, 인가: 방 출입권한) |