

# 2주차 과제 안내

## 1. 전반적인 규칙

- 주어진 엔드포인트를 Spring Boot로 구현하세요.
- 데이터는 HashMap<Long, Product>을 이용하여 메모리에 저장합니다.
- 각 클래스는 한 가지의 역할만을 수행합니다.
  - Repository: 데이터 CRUD와 관련된 역할을 수행합니다.
  - Controller: 요청을 핸들링하는 역할을 수행합니다.
- 이 과제는 테스트를 위한 HTML 파일이 제공됩니다.
  - `src/main/resources/static/` 경로에 제공된 파일을 넣으세요.
  - 모든 API를 올바르게 구현하여 제공된 HTML 페이지가 정상적으로 작동되는지 확인하세요.

## 2. 과제 명세

### 2.1. Domain

#### 2.1.1 Product

Product 클래스는 다음의 멤버변수와 각각의 getter, setter를 가집니다.

- id: Long
- name: String
- description: String
- price: Integer

### 2.2. Endpoint

#### GET | /v1/product

요청 바디는 필요하지 않습니다.

모든 상품을 반환합니다.

- Request Body: X
- Response Body: `List<Product>`

#### GET | /v1/product/{id}

요청 바디는 필요하지 않습니다.

PathVariable로 주어진 id의 상품을 반환합니다.

- Request Body: X
- Response Body: `Product`

## POST | /v1/product

상품을 등록합니다. ID는 요청 바디에 작성된 값과 관계없이 계속 증가합니다.  
요청 바디에는 등록할 Product가 제공됩니다.  
성공적으로 등록했다면 상품을 반환합니다.

- Request Body: `Product`
- Response Body: `Product`

## 3. 힌트

- 이 과제에서 사용되는 핵심 키워드는 다음과 같습니다
  - HTTP
    - GET
    - POST
  - `@RestController`
  - `@GetMapping`
  - `@PostMapping`
  - `@PathVariable`
  - 의존성 주입 (DI)

## 4. 테스트

- 아래의 페이지가 정상적으로 작동하는지 테스트해보세요.
  - <http://localhost:8080/product-add.html>
  - <http://localhost:8080/product-detail.html>