

mytheresa Promotions Test

Description

We want you to implement a REST API endpoint that given a list of products, applies some discounts to them and can be filtered.

You are free to choose whatever language and tools you are most comfortable with. Please add instructions on how to run it and publish it in Github.

What we expect

- Code structure/architecture must fit this use case, as simple or as complex needed to complete what is asked for.
- Test are a must. Code must be testable without requiring networking or the filesystem. Tests should be runnable with 1 command.
- The project must be runnable with 1 simple command from any machine.
- Explanations on decisions taken

Given this list of products:

```
{
  "products": [
    {
      "sku": "000001",
      "name": "BV Lean leather ankle boots",
      "category": "boots",
      "price": 89000
    },
    {
      "sku": "000002",
      "name": "BV Lean leather ankle boots",
      "category": "boots",
      "price": 99000
    },
    {
      "sku": "000003",
      "name": "Ashlington leather ankle boots",
      "category": "boots",
      "price": 71000
    },
    {
      "sku": "000004",
      "name": "Naima embellished suede sandals",
      "category": "sandals",
      "price": 79500
    },
    {
      "sku": "000005",
      "name": "Nathane leather sneakers",
      "category": "sneakers",
      "price": 59000
    }
  ]
}
```

You must take into account that this list could grow to have 20.000 products.

The prices are integers for example, 100.00€ would be 10000 .

You can store the products as you see fit (json file, in memory, rdbms of choice)

Given that:

- Products in the boots category have a 30% discount.
- The product with sku = 000003 has a 15% discount.
- When multiple discounts collide, the bigger discount must be applied.

Provide a single endpoint

GET /products

- Can be filtered by `category` as a query string parameter
- (*optional*) Can be filtered by `priceLessThan` as a query string parameter, this filter applies before discounts are applied and will show products with prices lesser than or equal the value provided.
- Returns a list of `Product` with the given discounts applied when necessary
- Must return at most 5 elements. (The order does not matter)

Product model

- `price.currency` is always `EUR`
- When a product does not have a discount, `price.final` and `price.original` should be the same number and `discount_percentage` should be null.
- When a product has a discount `price.original` is the original price, `price.final` is the amount with the discount applied and `discount_percentage` represents the applied discount with the `%` sign.

Example product with a discount of 30% applied.

```
{
  "sku": "000001",
  "name": "BV Lean leather ankle boots",
  "category": "boots",
  "price": {
    "original": 89000,
    "final": 62300,
    "discount_percentage": "30%",
    "currency": "EUR"
  }
}
```

Example product without a discount

```
{
  "sku": "000001",
  "name": "BV Lean leather ankle boots",
  "category": "boots",
  "price": {
    "original": 89000,
    "final": 89000,
    "discount_percentage": null,
    "currency": "EUR"
  }
}
```