1. Definición inicial del dominio

Para el comercio electrónico actual existe la tabla PRICES que refleja el precio a aplicar para un producto determinado de una marca concreta en un periodo de tiempo especificado.

Los campos que definen a la tabla PRICES son:

* BRAND\_ID: foreign key de la cadena del grupo (Ejemplo 1 = ZARA).
* START\_DATE: fecha de inicio en el que aplica el precio tarifa indicado.
* END\_DATE: fecha de fin en el que aplica el precio tarifa indicado.
* PRICE\_LIST: Identificador de la tarifa de precios aplicable.
* PRODUCT\_ID: Identificador del código de producto.
* PRIORITY: Desambiguador de aplicación de precios. Si dos tarifas coinciden en un rago de fechas se aplica la de mayor prioridad (mayor valor numérico).
* PRICE: precio final de venta.
* CURR: iso de la moneda.

1. Modelo del dominio

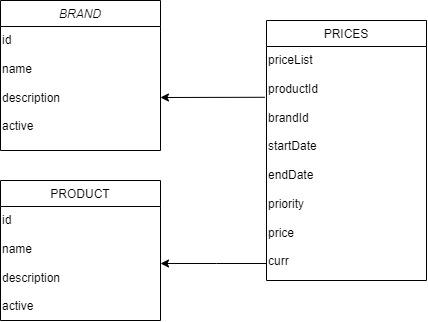
Con el propósito de representar las relaciones de la tarifa a aplicar para un producto durante un periodo de tiempo, se ha definido las tablas BRAND y PRODUCT.

Los campos que definen a la tabla BRAND son:

* id: Identificador del código de la marca.
* name: Nombre de la marca.
* description: Descripción de la marca.
* active: Indica si la marca se encuentra disponible.

Los campos que definen a la tabla PRODUCT son:

* id: Identificador del código del producto.
* name: Nombre del producto.
* description: Descripción del producto.
* active: Indica si el producto se encuentra disponible.



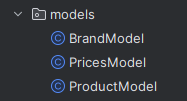
Para la tabla PRODUCT se define una relación de “1 a N” con la tabla PRICES porque para un producto puede existir diferentes precios a aplicar.

Para la tabla BRAND se define una relación de “1 a N” con la tabla PRICES porque para una marca puede existir diferentes relaciones de precio de sus productos a aplicar.

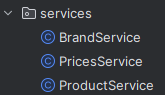
1. Implementación

3.1 Conceptos

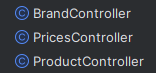
Se ha definido las clases de los 3 modelos con los atributos de acuerdo a la definición anterior.



Se ha definido los servicios con las funciones de acceso a la BBDD de los 3 modelos.



Se han generado los controladores REST para dar disponibilidad a los endpoints que efectuarán las consultas sobre la BBDD.



3.1 Servicios



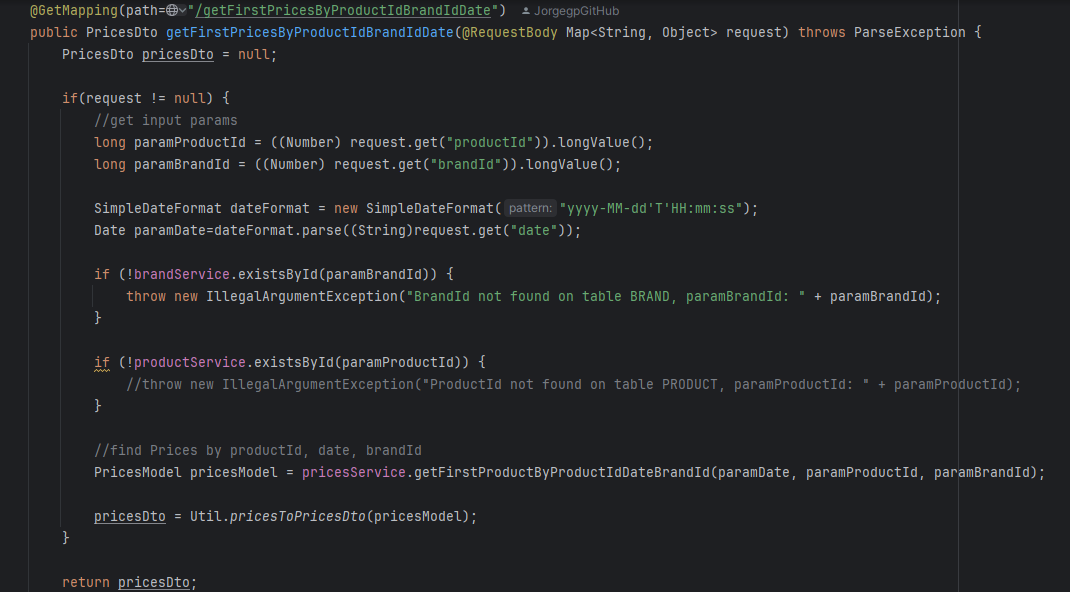
POST con path “/brand/registerData” que recibe en el body un JSON con la lista de marcas que se deben insertar en la tabla BRAND.



POST con path “/product/registerData” que recibe en el body un JSON con la lista de productos que se deben insertar en la tabla BRAND.



POST con path “/prices/registerData” que recibe en el body un JSON con la lista de precios a aplicar para cada producto de una marca en un periodo concreto.

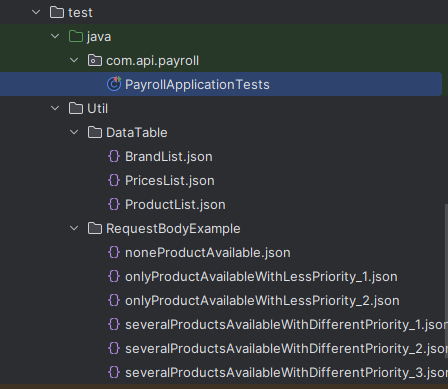


GET con path “/prices/getFirstPricesByProductIdBrandIdDate” que recibe en el body un JSON con un id de producto, id de marca y una fecha y busca en la tabla PRICES un registro que cumpla con las condiciones.

Nota: Se aplica la validación de los campos de entrada para comprobar si el id de la marca y el id del producto especificados existen.

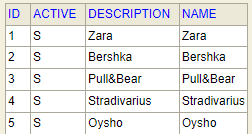
1. BATERÍA DE PRUEBAS

4.1 Tests unitario

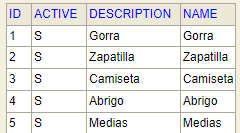


Se han definido los ficheros “BrandList.json”, “PricesList.json” y “ProductList.json” con los datos que serán registrados en la BBDD. De esta forma, cada vez que se ejecute un test, lo primero que se aplicará será la inserción de los datos.

RESULTADO TABLA BRAND



RESULTADO TABLA PRODUCT



RESULTADO TABLA PRICES



En el fichero “PayrollApplicationTests” se puede lanzar las pruebas solicitadas:

* Test “onlyProductAvailableWithLessPriority\_1”

Request body:

{

"brandId": 1,

"date": "2020-06-14T10:00:00",

"productId": 1

}

Encuentra resultado:

{

"productId": 1,

"priceList": 1,

"brandId": 1,

"price": 35.50,

"curr": "EUR",

"startDate": "2020-06-14T00:00:00.000+00:00",

"endDate": "2020-12-31T23:59:59.000+00:00"

},

* Test “onlyProductAvailableWithLessPriority\_2”

Request body:

{

"brandId": 1,

"date": "2020-06-14T21:00:00",

"productId": 1

}

Encuentra resultado:

{

"productId": 1,

"priceList": 1,

"brandId": 1,

"price": 35.50,

"curr": "EUR",

"startDate": "2020-06-14T00:00:00.000+00:00",

"endDate": "2020-12-31T23:59:59.000+00:00"

},

* Test “severalProductsAvailableWithDifferentPriority\_1”

Request body:

{

"brandId": 1,

"date": "2020-06-14T16:00:00",

"productId": 1

}

Encuentra resultado:

{

"productId": 1,

"priceList": 1,

"brandId": 1,

"price": 35.50,

"curr": "EUR",

"startDate": "2020-06-14T00:00:00.000+00:00",

"endDate": "2020-12-31T23:59:59.000+00:00"

},

* Test “severalProductsAvailableWithDifferentPriority\_2”

Request body:

{

"brandId": 1,

"date": "2020-06-15T10:00:00",

"productId": 1

}

Encuentra resultado:

{

"productId": 1,

"priceList": 3,

"brandId": 1,

"price": 30.50,

"curr": "EUR",

"startDate": "2020-06-15T00:00:00.000+00:00",

"endDate": "2020-06-15T11:00:00.000+00:00"

},

* Test “severalProductsAvailableWithDifferentPriority\_3”

Request body:

{

"brandId": 1,

"date": "2020-06-16T21:00:00",

"productId": 1

}

Encuentra resultado:

{

"productId": 1,

"priceList": 4,

"brandId": 1,

"price": 38.95,

"curr": "EUR",

"startDate": "2020-06-15T16:00:00.000+00:00",

"endDate": "2020-12-31T23:59:59.000+00:00"

}

* Test “noneProductAvailable"

{

"brandId": 1,

"date": "2010-06-16T21:00:00",

"productId": 1

}

No encuentra ningún resultado.

Resultado global de pruebas OK.



4.2 Test de integración

Se inicia la aplicación y se ejecuta la prueba.

