Explanation for testing (almost) each endpoint for the Accesa Internship Coding Challenge application

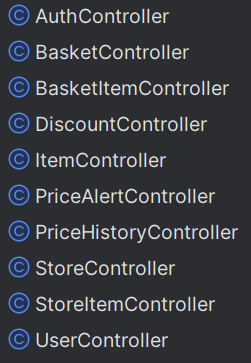
By Moldovan Ilie-Gabriel - [moldovan.gabi75@gmail.com](mailto:moldovan.gabi75@gmail.com)

So first I want to point out the fact that the tests are done on the dummy data that I provided.

If you want to test the application out for yourself, please check the Github repository to find out how you can do it:

<https://github.com/GabiMoldovan/Accesa-Coding-Challenge>

These are the controllers:

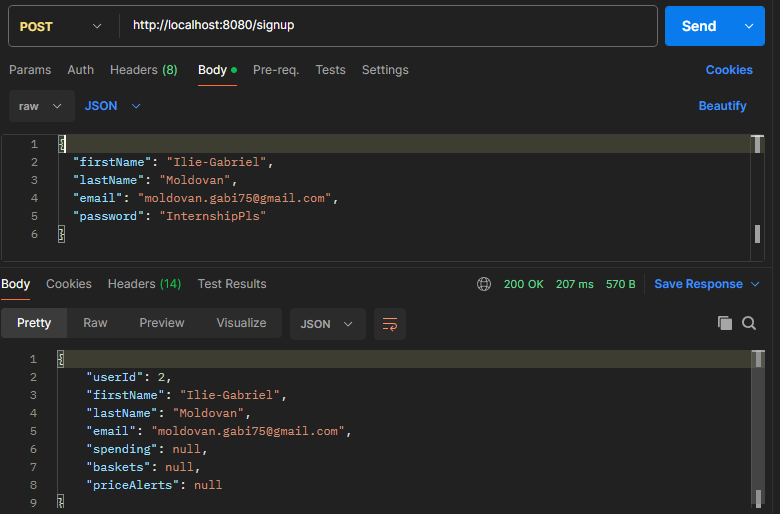


I’m using Postman to test them

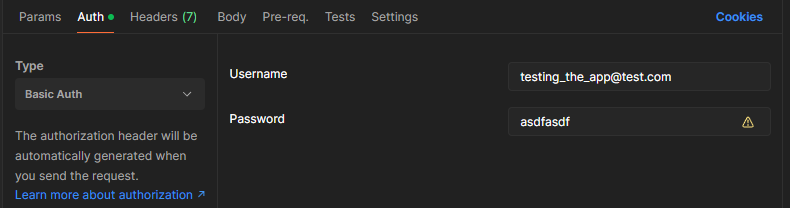
==========================================================

AuthController:

The endpoint for registering an account

POST http://localhost:8080/signup  
*Content-Type*: application/json  
*{  
 "firstName"*: "",  
 *"lastName"*: "",  
 *"email"*: "",  
 *"password"*: ""  
*}*

I will use the first user, since that is the one that’s populated with the data base initializer

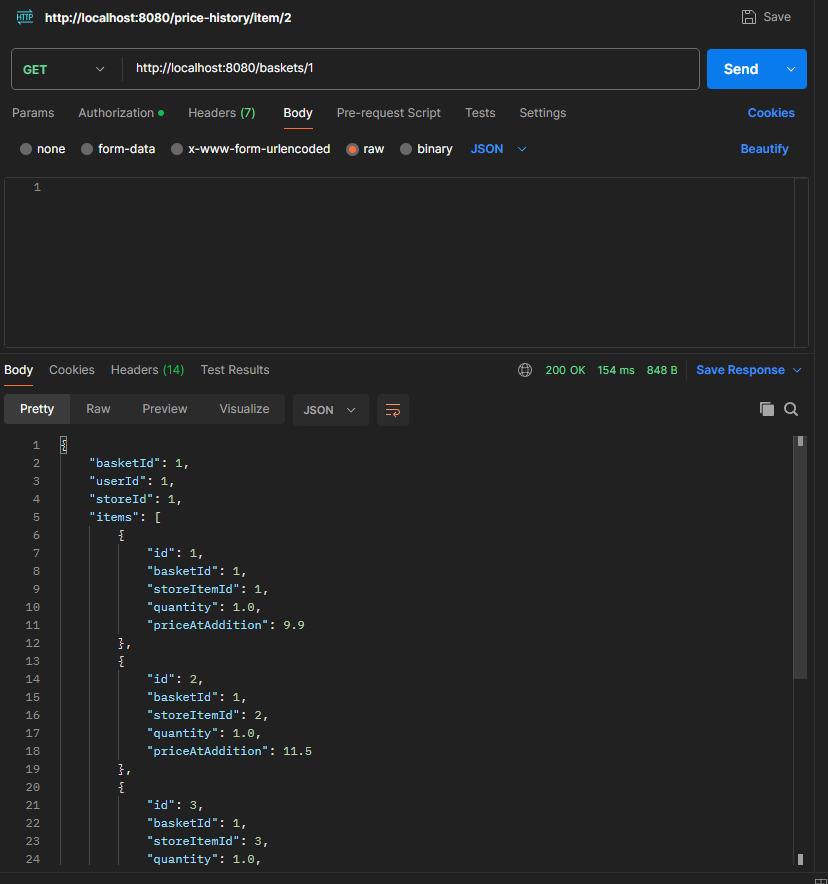
I will add the credentials of the account to test the rest of the endpoints

==========================================================

Basket Controller

The endpoint for getting a basket with a specific id

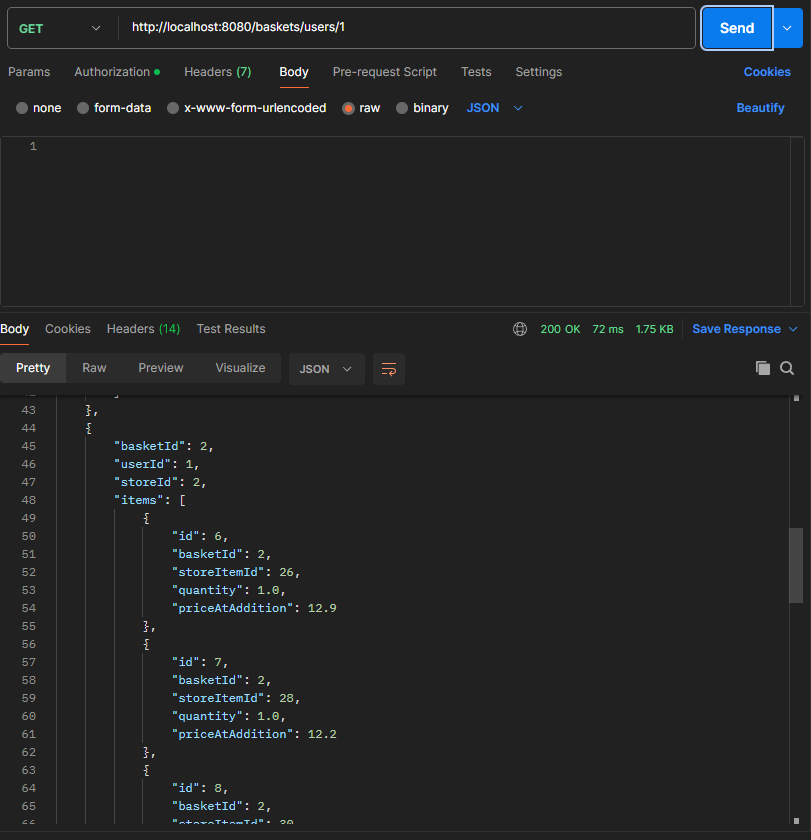
GET http://localhost:8080/baskets/{{basketId}}



==========================================================

The endpoint to get all the baskets that a user has

GET http://localhost:8080/baskets/users/{{userId}}



==========================================================

POST http://localhost:8080/baskets/{{userId}}/{{basketId}}/checkout

This endpoint is used to create a spending out of a basket

It is used when the user confirms the purchase of the items in the basket

This is basket nr 1:

{

        "basketId": 1,

        "userId": 1,

        "storeId": 1,

        "items": [

            {

                "id": 1,

                "basketId": 1,

                "storeItemId": 1,

                "quantity": 1.0,

                "priceAtAddition": 9.9

            },

            {

                "id": 2,

                "basketId": 1,

                "storeItemId": 2,

                "quantity": 1.0,

                "priceAtAddition": 11.5

            },

            {

                "id": 3,

                "basketId": 1,

                "storeItemId": 3,

                "quantity": 1.0,

                "priceAtAddition": 13.2

            },

            {

                "id": 4,

                "basketId": 1,

                "storeItemId": 4,

                "quantity": 1.0,

                "priceAtAddition": 12.8

            },

            {

                "id": 5,

                "basketId": 1,

                "storeItemId": 5,

                "quantity": 1.0,

                "priceAtAddition": 3.5

            }

        ]

    },

And I’ll make a checkout on it:



I get:

{

    "spendingId": 2,

    "userId": 1,

    "storeId": 1,

    "purchasedItems": [

        {

            "id": 5,

            "spendingId": 2,

            "itemName": "lapte zuzu",

            "pricePerUnit": 9.9,

            "units": 1.0,

            "unitType": "L"

        },

        {

            "id": 6,

            "spendingId": 2,

            "itemName": "iaurt grecesc",

            "pricePerUnit": 11.5,

            "units": 1.0,

            "unitType": "KG"

        },

        {

            "id": 7,

            "spendingId": 2,

            "itemName": "oua marimea M",

            "pricePerUnit": 13.2,

            "units": 1.0,

            "unitType": "BUC"

        },

        {

            "id": 8,

            "spendingId": 2,

            "itemName": "branza telemea",

            "pricePerUnit": 12.8,

            "units": 1.0,

            "unitType": "KG"

        },

        {

            "id": 9,

            "spendingId": 2,

            "itemName": "paine alba",

            "pricePerUnit": 3.5,

            "units": 1.0,

            "unitType": "G"

        }

    ],

    "totalPrice": 50.9,

    "purchaseDate": "2025-05-13T23:51:27.5788344"

}

==========================================================

I reset the data in the data base so I can have the initial data again.

I created the user again:

{

    "userId": 1,

    "firstName": "MyTestingUser",

    "lastName": "AccesaInternship",

    "email": "testing\_the\_app@test.com",

    "spending": null,

    "baskets": null,

    "priceAlerts": null

}

And I initialized the data base using the DatabaseInitializer that I provided

==========================================================

POST http://localhost:8080/baskets/users/{{userId}}/optimize

This endpoint is used to solve one of the requirements:

Daily Shopping Basket Monitoring: - Help users split their basket into shopping lists that optimise for cost savings

I solved it by letting users pick what items they want, and added a "optimization" endpoint that works like this:

Let's say that the user has 3 baskets, each basket for one store, with their specific items in it.

By accessing the endpoint, the backend gets the pool of items that's in all the baskets, and for each item, we search

the stores to find the cheapest possible instance of that item. We do that for all the items

So in the end, the user will have their baskets shuffled, but the baskets will contain the cheapest items in the mall.

These are the baskets before the optimization:

{

    "basketId": 1,

    "userId": 1,

    "storeId": 1,

    "items": [

        {

            "id": 1,

            "basketId": 1,

            "storeItemId": 1,

            "quantity": 1.0,

            "priceAtAddition": 9.9

        },

        {

            "id": 2,

            "basketId": 1,

            "storeItemId": 2,

            "quantity": 1.0,

            "priceAtAddition": 11.5

        },

        {

            "id": 3,

            "basketId": 1,

            "storeItemId": 3,

            "quantity": 1.0,

            "priceAtAddition": 13.2

        },

        {

            "id": 4,

            "basketId": 1,

            "storeItemId": 4,

            "quantity": 1.0,

            "priceAtAddition": 12.8

        },

        {

            "id": 5,

            "basketId": 1,

            "storeItemId": 5,

            "quantity": 1.0,

            "priceAtAddition": 3.5

        }

    ]

}

And these are after the optimization:  
  
[

    {

        "basketId": 1,

        "userId": 1,

        "storeId": 1,

        "items": [

            {

                "id": 1,

                "basketId": 1,

                "storeItemId": 1,

                "quantity": 2.0,

                "priceAtAddition": 9.9

            },

            {

                "id": 2,

                "basketId": 1,

                "storeItemId": 2,

                "quantity": 1.0,

                "priceAtAddition": 11.5

            },

            {

                "id": 4,

                "basketId": 1,

                "storeItemId": 4,

                "quantity": 2.0,

                "priceAtAddition": 12.8

            },

            {

                "id": 5,

                "basketId": 1,

                "storeItemId": 5,

                "quantity": 3.0,

                "priceAtAddition": 3.5

            },

            {

                "id": 17,

                "basketId": 1,

                "storeItemId": 9,

                "quantity": 1.0,

                "priceAtAddition": 4.4

            },

            {

                "id": 18,

                "basketId": 1,

                "storeItemId": 11,

                "quantity": 1.0,

                "priceAtAddition": 6.1

            },

            {

                "id": 19,

                "basketId": 1,

                "storeItemId": 13,

                "quantity": 1.0,

                "priceAtAddition": 7.1

            },

            {

                "id": 20,

                "basketId": 1,

                "storeItemId": 6,

                "quantity": 1.0,

                "priceAtAddition": 6.8

            }

        ]

    },

    {

        "basketId": 2,

        "userId": 1,

        "storeId": 2,

        "items": [

            {

                "id": 7,

                "basketId": 2,

                "storeItemId": 28,

                "quantity": 2.0,

                "priceAtAddition": 12.2

            },

            {

                "id": 9,

                "basketId": 2,

                "storeItemId": 32,

                "quantity": 2.0,

                "priceAtAddition": 26.5

            }

        ]

    },

    {

        "basketId": 3,

        "userId": 1,

        "storeId": 3,

        "items": []

    }

]

I suggest that you compare the new result to the data in the dummy data folder. You can see that the items remain exactly the same, but we have the items from other stores that are the cheapest. But still the same items.

==========================================================

Resetting the data to the initial data using the databaseinitializer so that we have the initial on our account again

==========================================================

GET http://localhost:8080/baskets/users/{{userId}}/stores/{{storeId}}

This endpoint is used to get a specific basket of a specific user

I made a request for storeId 3 and user 1 to get the user’s basket at store with id 3

{

    "basketId": 3,

    "userId": 1,

    "storeId": 3,

    "items": [

        {

            "id": 13,

            "basketId": 3,

            "storeItemId": 50,

            "quantity": 1.0,

            "priceAtAddition": 13.1

        },

        {

            "id": 14,

            "basketId": 3,

            "storeItemId": 51,

            "quantity": 1.0,

            "priceAtAddition": 3.6

        },

        {

            "id": 15,

            "basketId": 3,

            "storeItemId": 52,

            "quantity": 1.0,

            "priceAtAddition": 7.0

        },

        {

            "id": 16,

            "basketId": 3,

            "storeItemId": 53,

            "quantity": 1.0,

            "priceAtAddition": 27.9

        }

    ]

}

==========================================================

POST http://localhost:8080/basket-items?  
 userId={{$random.integer(100)}}&  
 storeId={{$random.integer(100)}}  
 Content-Type: application/json

This endpoint is used to add an item into a user’s basket

I entered this url:

http://localhost:8080/basket-items?userId=1&storeId=3

With body

{

    "basketId": 3,

    "storeItemId": 5,

    "quantity": 10

}

And received:

{

    "id": 17,

    "basketId": 3,

    "storeItemId": 5,

    "quantity": 10.0,

    "priceAtAddition": 3.5

}

==========================================================

GET http://localhost:8080/basket-items/{{id}}

This endpoint is used to get a specific item from a user basket

For this url:

http://localhost:8080/basket-items/11

I received:

{

    "id": 11,

    "basketId": 2,

    "storeItemId": 36,

    "quantity": 1.0,

    "priceAtAddition": 6.1

}

==========================================================

DELETE http://localhost:8080/basket-items/{{id}}

This endpoint is used to delete an item from someone’s basket

Let’s delete the item that we previously added:  
So for this url: http://localhost:8080/basket-items/17

I received this:

{

    "id": 17,

    "basketId": 3,

    "storeItemId": 5,

    "quantity": 10.0,

    "priceAtAddition": 3.5

}

It is indeed the item that we added into our basket

==========================================================

PATCH http://localhost:8080/basket-items/{{id}}  
*Content-Type*: application/json  
  
{  
 "quantity": 0,  
}

This endpoint is used to update the quantity of an item in a basket

I updated the quantity of the basketItem 16:

http://localhost:8080/basket-items/16

And received:

{

    "id": 16,

    "basketId": 3,

    "storeItemId": 53,

    "quantity": 7.0,

    "priceAtAddition": 27.9

}

==========================================================

POST http://localhost:8080/items  
*Content-Type*: application/json

This endpoint is used to create a new item

I used the body:

{

  "itemName": "Meteoritzi",

  "category": "GUSTARI",

  "brand": "Romanesc",

  "unitType": "G"

}

And got:

{

    "itemId": 31,

    "itemName": "Meteoritzi",

    "category": "GUSTARI",

    "brand": "Romanesc",

    "unitType": "G"

}

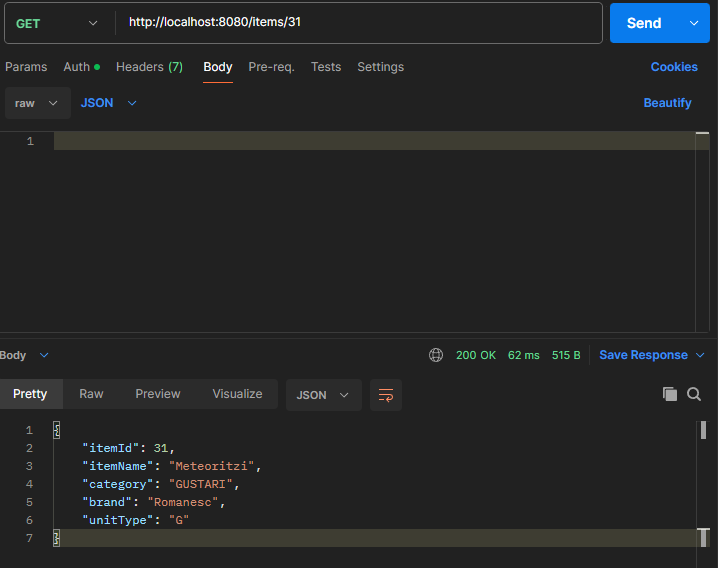
==========================================================

GET http://localhost:8080/items/{{id}}  
*Content-Type*: application/json  
  
{  
 "itemName": "",  
 "category": "LACTATE",  
 "brand": "",  
 "unitType": "L"  
}

This endpoint is used to retrieve a specific item by their id

I’ll make a request for the item that I just created:

I attached a photo to see that it works for this one



==========================================================

PATCH http://localhost:8080/items/{{id}}  
*Content-Type*: application/json  
  
{  
 "itemName": "",  
 "category": "LACTATE",  
 "brand": "",  
 "unitType": "L"  
}

This endpoint is used to update an item

I updated the same item using http://localhost:8080/items/31

And the body:

{

    "itemName": "Meteoriti intergalactici",

    "category": "GUSTARI",

    "brand": "Romanesc",

    "unitType": "G"

}

And I received this:

{

    "itemId": 31,

    "itemName": "Meteoriti intergalactici",

    "category": "GUSTARI",

    "brand": "Romanesc",

    "unitType": "G"

}

==========================================================

DELETE http://localhost:8080/items/{{id}}  
*Content-Type*: application/json  
  
{  
 "itemName": "",  
 "category": "LACTATE",  
 "brand": "",  
 "unitType": "L"  
}

This endpoint is used to delete an item

I’ll delete the same item that I created before: http://localhost:8080/items/31

And I got:

{

    "itemId": 31,

    "itemName": "Meteoriti intergalactici",

    "category": "GUSTARI",

    "brand": "Romanesc",

    "unitType": "G"

}

==========================================================

GET http://localhost:8080/items  
*Content-Type*: application/json  
  
{  
 "itemName": "",  
 "category": "LACTATE",  
 "brand": "",  
 "unitType": "L"  
}

This endpoint is used to retrieve all the available items

I received:

[

    {

        "itemId": 1,

        "itemName": "lapte zuzu",

        "category": "LACTATE",

        "brand": "Zuzu",

        "unitType": "L"

    },

    {

        "itemId": 2,

        "itemName": "iaurt grecesc",

        "category": "LACTATE",

        "brand": "Proxi",

        "unitType": "KG"

    },

    {

        "itemId": 3,

        "itemName": "oua marimea M",

        "category": "OUA",

        "brand": "facute de cocos",

        "unitType": "BUC"

    },

    {

        "itemId": 4,

        "itemName": "branza telemea",

        "category": "LACTATE",

        "brand": "Proxi",

        "unitType": "KG"

    },

    {

        "itemId": 5,

        "itemName": "paine alba",

        "category": "PANIFICATIE",

        "brand": "Proxi",

        "unitType": "G"

    },

    {

        "itemId": 6,

        "itemName": "rosii cherry",

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic",

        "unitType": "G"

    },

    {

        "itemId": 7,

        "itemName": "piept pui",

        "category": "CARNE",

        "brand": "Avicola",

        "unitType": "KG"

    },

    {

        "itemId": 8,

        "itemName": "spaghetti",

        "category": "PASTE\_FAINOASE",

        "brand": "Barilla",

        "unitType": "G"

    },

    {

        "itemId": 9,

        "itemName": "zahar tos",

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi",

        "unitType": "KG"

    },

    {

        "itemId": 10,

        "itemName": "apa plata",

        "category": "BAUTURI",

        "brand": "Aqua Carpatica",

        "unitType": "L"

    },

    {

        "itemId": 11,

        "itemName": "banane",

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic",

        "unitType": "KG"

    },

    {

        "itemId": 12,

        "itemName": "ulei floarea-soarelui",

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Spornic",

        "unitType": "L"

    },

    {

        "itemId": 13,

        "itemName": "biscuiti cu unt",

        "category": "GUSTARI",

        "brand": "RoStar",

        "unitType": "KG"

    },

    {

        "itemId": 14,

        "itemName": "cafea macinata",

        "category": "CAFEA",

        "brand": "Tchibo",

        "unitType": "KG"

    },

    {

        "itemId": 15,

        "itemName": "detergent lichid",

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Dero",

        "unitType": "L"

    },

    {

        "itemId": 16,

        "itemName": "sampon par gras",

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Schauma",

        "unitType": "ML"

    },

    {

        "itemId": 17,

        "itemName": "hartie igienica",

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Motto",

        "unitType": "ROLE"

    },

    {

        "itemId": 18,

        "itemName": "piper negru macinat",

        "category": "CONDIMENTE",

        "brand": "Kamis",

        "unitType": "G"

    },

    {

        "itemId": 19,

        "itemName": "vin alb demisec",

        "category": "BAUTURI",

        "brand": "Cotnari",

        "unitType": "L"

    },

    {

        "itemId": 20,

        "itemName": "ciocolata neagra 70%",

        "category": "GUSTARI",

        "brand": "Poiana",

        "unitType": "G"

    },

    {

        "itemId": 21,

        "itemName": "suc pepsi",

        "category": "BAUTURI",

        "brand": "Pepsi",

        "unitType": "L"

    },

    {

        "itemId": 22,

        "itemName": "biscuiti oreo",

        "category": "GUSTARI",

        "brand": "Oreo",

        "unitType": "G"

    },

    {

        "itemId": 23,

        "itemName": "ceapa galbena",

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic",

        "unitType": "KG"

    },

    {

        "itemId": 24,

        "itemName": "cartofi albi",

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic",

        "unitType": "KG"

    },

    {

        "itemId": 25,

        "itemName": "file somon",

        "category": "PESTE",

        "brand": "Lidl",

        "unitType": "KG"

    },

    {

        "itemId": 26,

        "itemName": "crema de branza",

        "category": "LACTATE",

        "brand": "Almette",

        "unitType": "KG"

    },

    {

        "itemId": 27,

        "itemName": "malai extra",

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Pambac",

        "unitType": "KG"

    },

    {

        "itemId": 28,

        "itemName": "orez bob lung",

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Deroni",

        "unitType": "KG"

    },

    {

        "itemId": 29,

        "itemName": "pasta de dinti",

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Colgate",

        "unitType": "ML"

    },

    {

        "itemId": 30,

        "itemName": "morcovi",

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic",

        "unitType": "KG"

    }

]

==========================================================

###  
POST http://localhost:8080/stores  
*Content-Type*: application/json  
  
{  
 "companyName": "",  
 "storeItems": [],  
 "discounts": [],  
 "priceHistory": []  
}

This endpoint is used to create a new store in the mall

I used the url http://localhost:8080/stores

And body

{

  "companyName": "CatStore",

  "storeItems": [],

  "discounts": [],

  "priceHistory": []

}

To create a new store, and received the response:

{

    "storeId": 4,

    "companyName": "CatStore",

    "storeItems": [],

    "discounts": [],

    "priceHistory": []

}

==========================================================

GET http://localhost:8080/stores/{{id}}  
*Content-Type*: application/json  
  
{  
 "companyName": ""  
}

This endpoint is used to retrieve a store by id

I used the url: http://localhost:8080/stores/4

And I received:

{

    "storeId": 4,

    "companyName": "CatStore",

    "storeItems": [],

    "discounts": [],

    "priceHistory": []

}

==========================================================

###  
PATCH http://localhost:8080/stores/{{id}}  
*Content-Type*: application/json

This endpoint is used to update the name of the store

I used the url: http://localhost:8080/stores/4

With the body:

{

    "companyName": "It is now a dog store"

}

And received the response:

{

    "storeId": 4,

    "companyName": "It is now a dog store",

    "storeItems": [],

    "discounts": [],

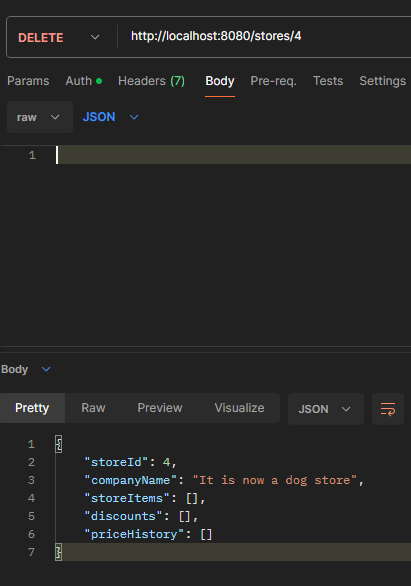
    "priceHistory": []

}

==========================================================

DELETE http://localhost:8080/stores/{{id}}  
*Content-Type*: application/json  
  
{  
 "companyName": ""  
}

This endpoint is used to delete a store



==========================================================

###  
GET http://localhost:8080/price-history/dynamic-price-history?  
 storeId={{$random.integer(100)}}&  
 category={{$random.alphanumeric(8)}}&  
 brand={{$random.alphanumeric(8)}}

This endpoint is used to solve one of the requirements:

Dynamic Price History Graphs: - Provide data points that would allow a frontend to calculate and display price trends over time for individual products.  
- This data should be filterable by store, product category, or brand.

If I use this url: http://localhost:8080/price-history/dynamic-price-history

I receive:

[

    {

        "priceHistoryId": 1,

        "storeItemId": 1,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.564728",

        "price": 9.9,

        "category": "LACTATE",

        "brand": "Zuzu"

    },

    {

        "priceHistoryId": 2,

        "storeItemId": 2,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.57073",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 3,

        "storeItemId": 3,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.57373",

        "price": 13.2,

        "category": "OUA",

        "brand": "facute de cocos"

    },

    {

        "priceHistoryId": 4,

        "storeItemId": 4,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.575728",

        "price": 12.8,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 5,

        "storeItemId": 5,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.577728",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 6,

        "storeItemId": 6,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.580728",

        "price": 6.8,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 7,

        "storeItemId": 7,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.582728",

        "price": 28.5,

        "category": "CARNE",

        "brand": "Avicola"

    },

    {

        "priceHistoryId": 8,

        "storeItemId": 8,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.58473",

        "price": 5.8,

        "category": "PASTE\_FAINOASE",

        "brand": "Barilla"

    },

    {

        "priceHistoryId": 9,

        "storeItemId": 9,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.586731",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 10,

        "storeItemId": 10,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.587728",

        "price": 5.2,

        "category": "BAUTURI",

        "brand": "Aqua Carpatica"

    },

    {

        "priceHistoryId": 11,

        "storeItemId": 11,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.589729",

        "price": 6.1,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 12,

        "storeItemId": 12,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.592728",

        "price": 9.2,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Spornic"

    },

    {

        "priceHistoryId": 13,

        "storeItemId": 13,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.593728",

        "price": 7.1,

        "category": "GUSTARI",

        "brand": "RoStar"

    },

    {

        "priceHistoryId": 14,

        "storeItemId": 14,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.596728",

        "price": 22.4,

        "category": "CAFEA",

        "brand": "Tchibo"

    },

    {

        "priceHistoryId": 15,

        "storeItemId": 15,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.598728",

        "price": 49.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Dero"

    },

    {

        "priceHistoryId": 16,

        "storeItemId": 16,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.600729",

        "price": 17.8,

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Schauma"

    },

    {

        "priceHistoryId": 17,

        "storeItemId": 17,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.601729",

        "price": 18.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Motto"

    },

    {

        "priceHistoryId": 18,

        "storeItemId": 18,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.603728",

        "price": 6.0,

        "category": "CONDIMENTE",

        "brand": "Kamis"

    },

    {

        "priceHistoryId": 19,

        "storeItemId": 19,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.604728",

        "price": 23.5,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 20,

        "storeItemId": 20,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.606728",

        "price": 3.9,

        "category": "GUSTARI",

        "brand": "Poiana"

    },

    {

        "priceHistoryId": 21,

        "storeItemId": 21,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.607729",

        "price": 13.99,

        "category": "BAUTURI",

        "brand": "Pepsi"

    },

    {

        "priceHistoryId": 22,

        "storeItemId": 22,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.609728",

        "price": 3.49,

        "category": "GUSTARI",

        "brand": "Oreo"

    },

    {

        "priceHistoryId": 23,

        "storeItemId": 23,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.610728",

        "price": 2.95,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 24,

        "storeItemId": 24,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.611728",

        "price": 3.1,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 25,

        "storeItemId": 25,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.613729",

        "price": 22.0,

        "category": "PESTE",

        "brand": "Lidl"

    },

    {

        "priceHistoryId": 26,

        "storeItemId": 26,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.614728",

        "price": 12.9,

        "category": "LACTATE",

        "brand": "Zuzu"

    },

    {

        "priceHistoryId": 27,

        "storeItemId": 27,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.616728",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 28,

        "storeItemId": 28,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.617729",

        "price": 12.2,

        "category": "OUA",

        "brand": "facute de cocos"

    },

    {

        "priceHistoryId": 29,

        "storeItemId": 29,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.619729",

        "price": 13.1,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 30,

        "storeItemId": 30,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.621284",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 31,

        "storeItemId": 31,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62329",

        "price": 8.8,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 32,

        "storeItemId": 32,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.624293",

        "price": 26.5,

        "category": "CARNE",

        "brand": "Avicola"

    },

    {

        "priceHistoryId": 33,

        "storeItemId": 33,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.626292",

        "price": 5.8,

        "category": "PASTE\_FAINOASE",

        "brand": "Barilla"

    },

    {

        "priceHistoryId": 34,

        "storeItemId": 34,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.627291",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 35,

        "storeItemId": 35,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62829",

        "price": 5.2,

        "category": "BAUTURI",

        "brand": "Aqua Carpatica"

    },

    {

        "priceHistoryId": 36,

        "storeItemId": 36,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62929",

        "price": 6.1,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 37,

        "storeItemId": 37,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63129",

        "price": 9.2,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Spornic"

    },

    {

        "priceHistoryId": 38,

        "storeItemId": 38,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63329",

        "price": 7.1,

        "category": "GUSTARI",

        "brand": "RoStar"

    },

    {

        "priceHistoryId": 39,

        "storeItemId": 39,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63429",

        "price": 22.4,

        "category": "CAFEA",

        "brand": "Tchibo"

    },

    {

        "priceHistoryId": 40,

        "storeItemId": 40,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63529",

        "price": 49.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Dero"

    },

    {

        "priceHistoryId": 41,

        "storeItemId": 41,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.637291",

        "price": 17.8,

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Schauma"

    },

    {

        "priceHistoryId": 42,

        "storeItemId": 42,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.638289",

        "price": 18.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Motto"

    },

    {

        "priceHistoryId": 43,

        "storeItemId": 43,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.640291",

        "price": 6.0,

        "category": "CONDIMENTE",

        "brand": "Kamis"

    },

    {

        "priceHistoryId": 44,

        "storeItemId": 44,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.641291",

        "price": 23.5,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 45,

        "storeItemId": 45,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.64229",

        "price": 3.9,

        "category": "GUSTARI",

        "brand": "Poiana"

    },

    {

        "priceHistoryId": 46,

        "storeItemId": 46,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.644291",

        "price": 14.5,

        "category": "BAUTURI",

        "brand": "Pepsi"

    },

    {

        "priceHistoryId": 47,

        "storeItemId": 47,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.645289",

        "price": 10.1,

        "category": "LACTATE",

        "brand": "Zuzu"

    },

    {

        "priceHistoryId": 48,

        "storeItemId": 48,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.647289",

        "price": 11.8,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 49,

        "storeItemId": 49,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.64829",

        "price": 13.5,

        "category": "OUA",

        "brand": "facute de cocos"

    },

    {

        "priceHistoryId": 50,

        "storeItemId": 50,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65029",

        "price": 13.1,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 51,

        "storeItemId": 51,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.651289",

        "price": 3.6,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 52,

        "storeItemId": 52,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65329",

        "price": 7.0,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 53,

        "storeItemId": 53,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65429",

        "price": 27.9,

        "category": "CARNE",

        "brand": "Avicola"

    },

    {

        "priceHistoryId": 54,

        "storeItemId": 54,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.656291",

        "price": 5.9,

        "category": "PASTE\_FAINOASE",

        "brand": "Barilla"

    },

    {

        "priceHistoryId": 55,

        "storeItemId": 55,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65729",

        "price": 4.5,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 56,

        "storeItemId": 56,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.658289",

        "price": 5.3,

        "category": "BAUTURI",

        "brand": "Aqua Carpatica"

    },

    {

        "priceHistoryId": 57,

        "storeItemId": 57,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.660289",

        "price": 6.2,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 58,

        "storeItemId": 58,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.662289",

        "price": 9.5,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Spornic"

    },

    {

        "priceHistoryId": 59,

        "storeItemId": 59,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.663289",

        "price": 7.5,

        "category": "GUSTARI",

        "brand": "RoStar"

    },

    {

        "priceHistoryId": 60,

        "storeItemId": 60,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.66429",

        "price": 23.0,

        "category": "CAFEA",

        "brand": "Tchibo"

    },

    {

        "priceHistoryId": 61,

        "storeItemId": 61,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.666291",

        "price": 50.5,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Dero"

    },

    {

        "priceHistoryId": 62,

        "storeItemId": 62,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.667289",

        "price": 18.0,

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Schauma"

    },

    {

        "priceHistoryId": 63,

        "storeItemId": 63,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.668291",

        "price": 19.2,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Motto"

    },

    {

        "priceHistoryId": 64,

        "storeItemId": 64,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.67129",

        "price": 6.1,

        "category": "CONDIMENTE",

        "brand": "Kamis"

    },

    {

        "priceHistoryId": 65,

        "storeItemId": 65,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.672291",

        "price": 24.0,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 66,

        "storeItemId": 66,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.673291",

        "price": 4.1,

        "category": "GUSTARI",

        "brand": "Poiana"

    },

    {

        "priceHistoryId": 67,

        "storeItemId": 67,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.675289",

        "price": 13.99,

        "category": "BAUTURI",

        "brand": "Pepsi"

    },

    {

        "priceHistoryId": 68,

        "storeItemId": 68,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.67629",

        "price": 3.99,

        "category": "GUSTARI",

        "brand": "Oreo"

    },

    {

        "priceHistoryId": 69,

        "storeItemId": 69,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.677289",

        "price": 3.1,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 70,

        "storeItemId": 70,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.679291",

        "price": 2.99,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 71,

        "storeItemId": 71,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.680291",

        "price": 24.5,

        "category": "PESTE",

        "brand": "Lidl"

    },

    {

        "priceHistoryId": 72,

        "storeItemId": 72,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.68129",

        "price": 8.5,

        "category": "LACTATE",

        "brand": "Almette"

    },

    {

        "priceHistoryId": 73,

        "storeItemId": 73,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.68329",

        "price": 5.5,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Pambac"

    },

    {

        "priceHistoryId": 74,

        "storeItemId": 74,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.684291",

        "price": 8.0,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Deroni"

    },

    {

        "priceHistoryId": 75,

        "storeItemId": 75,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.686291",

        "price": 10.5,

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Colgate"

    },

    {

        "priceHistoryId": 76,

        "storeItemId": 76,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.687291",

        "price": 2.6,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 77,

        "storeItemId": 19,

        "storeId": 1,

        "date": "2025-05-14T00:09:00.048876",

        "price": 21.15,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 78,

        "storeItemId": 20,

        "storeId": 1,

        "date": "2025-05-14T00:09:00.059876",

        "price": 3.7050002,

        "category": "GUSTARI",

        "brand": "Poiana"

    }

]

If I use this url: http://localhost:8080/price-history/dynamic-price-history?storeId=2

I receive:

[

    {

        "priceHistoryId": 26,

        "storeItemId": 26,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.614728",

        "price": 12.9,

        "category": "LACTATE",

        "brand": "Zuzu"

    },

    {

        "priceHistoryId": 27,

        "storeItemId": 27,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.616728",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 28,

        "storeItemId": 28,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.617729",

        "price": 12.2,

        "category": "OUA",

        "brand": "facute de cocos"

    },

    {

        "priceHistoryId": 29,

        "storeItemId": 29,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.619729",

        "price": 13.1,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 30,

        "storeItemId": 30,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.621284",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 31,

        "storeItemId": 31,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62329",

        "price": 8.8,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 32,

        "storeItemId": 32,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.624293",

        "price": 26.5,

        "category": "CARNE",

        "brand": "Avicola"

    },

    {

        "priceHistoryId": 33,

        "storeItemId": 33,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.626292",

        "price": 5.8,

        "category": "PASTE\_FAINOASE",

        "brand": "Barilla"

    },

    {

        "priceHistoryId": 34,

        "storeItemId": 34,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.627291",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 35,

        "storeItemId": 35,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62829",

        "price": 5.2,

        "category": "BAUTURI",

        "brand": "Aqua Carpatica"

    },

    {

        "priceHistoryId": 36,

        "storeItemId": 36,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.62929",

        "price": 6.1,

        "category": "LEGUME\_SI\_FRUCTE",

        "brand": "Generic"

    },

    {

        "priceHistoryId": 37,

        "storeItemId": 37,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63129",

        "price": 9.2,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Spornic"

    },

    {

        "priceHistoryId": 38,

        "storeItemId": 38,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63329",

        "price": 7.1,

        "category": "GUSTARI",

        "brand": "RoStar"

    },

    {

        "priceHistoryId": 39,

        "storeItemId": 39,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63429",

        "price": 22.4,

        "category": "CAFEA",

        "brand": "Tchibo"

    },

    {

        "priceHistoryId": 40,

        "storeItemId": 40,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.63529",

        "price": 49.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Dero"

    },

    {

        "priceHistoryId": 41,

        "storeItemId": 41,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.637291",

        "price": 17.8,

        "category": "INGRIJIRE\_PERSONALA",

        "brand": "Schauma"

    },

    {

        "priceHistoryId": 42,

        "storeItemId": 42,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.638289",

        "price": 18.9,

        "category": "PRODUSE\_DE\_MENAJ",

        "brand": "Motto"

    },

    {

        "priceHistoryId": 43,

        "storeItemId": 43,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.640291",

        "price": 6.0,

        "category": "CONDIMENTE",

        "brand": "Kamis"

    },

    {

        "priceHistoryId": 44,

        "storeItemId": 44,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.641291",

        "price": 23.5,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 45,

        "storeItemId": 45,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.64229",

        "price": 3.9,

        "category": "GUSTARI",

        "brand": "Poiana"

    },

    {

        "priceHistoryId": 46,

        "storeItemId": 46,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.644291",

        "price": 14.5,

        "category": "BAUTURI",

        "brand": "Pepsi"

    }

]

If I use this url: http://localhost:8080/price-history/dynamic-price-history?brand=Proxi

To retrieve all price history of items that their brand is Proxi, I get this:

[

    {

        "priceHistoryId": 2,

        "storeItemId": 2,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.57073",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 4,

        "storeItemId": 4,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.575728",

        "price": 12.8,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 5,

        "storeItemId": 5,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.577728",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 9,

        "storeItemId": 9,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.586731",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 27,

        "storeItemId": 27,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.616728",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 29,

        "storeItemId": 29,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.619729",

        "price": 13.1,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 30,

        "storeItemId": 30,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.621284",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 34,

        "storeItemId": 34,

        "storeId": 2,

        "date": "2025-05-14T00:07:42.627291",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 48,

        "storeItemId": 48,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.647289",

        "price": 11.8,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 50,

        "storeItemId": 50,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65029",

        "price": 13.1,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 51,

        "storeItemId": 51,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.651289",

        "price": 3.6,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 55,

        "storeItemId": 55,

        "storeId": 3,

        "date": "2025-05-14T00:07:42.65729",

        "price": 4.5,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    }

]

If I combine them, to apply multiple filters:

http://localhost:8080/price-history/dynamic-price-history?storeId=1&brand=Proxi

I get this:

[

    {

        "priceHistoryId": 2,

        "storeItemId": 2,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.57073",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 4,

        "storeItemId": 4,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.575728",

        "price": 12.8,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 5,

        "storeItemId": 5,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.577728",

        "price": 3.5,

        "category": "PANIFICATIE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 9,

        "storeItemId": 9,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.586731",

        "price": 4.4,

        "category": "ALIMENTE\_DE\_BAZA",

        "brand": "Proxi"

    }

]

As you can see, the endpoint works even if you apply multiple filters.

Let’s try applying all 3 filters :D

http://localhost:8080/price-history/dynamic-price-history?storeId=1&category=LACTATE&brand=Proxi

We get:

[

    {

        "priceHistoryId": 2,

        "storeItemId": 2,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.57073",

        "price": 11.5,

        "category": "LACTATE",

        "brand": "Proxi"

    },

    {

        "priceHistoryId": 4,

        "storeItemId": 4,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.575728",

        "price": 12.8,

        "category": "LACTATE",

        "brand": "Proxi"

    }

]

==========================================================

GET http://localhost:8080/price-history/item/{{itemId}}/store/{{storeId}}

This endpoint is used to get all the price history for an item at a specific store

I used the url: http://localhost:8080/price-history/item/19/store/1

And received:

[

    {

        "priceHistoryId": 19,

        "storeItemId": 19,

        "storeId": 1,

        "date": "2025-05-14T00:07:42.604728",

        "price": 23.5,

        "category": "BAUTURI",

        "brand": "Cotnari"

    },

    {

        "priceHistoryId": 77,

        "storeItemId": 19,

        "storeId": 1,

        "date": "2025-05-14T00:09:00.048876",

        "price": 21.15,

        "category": "BAUTURI",

        "brand": "Cotnari"

    }

]

The reason why we have 2 entries is because the storeItem has a discount applied

So the first entry is when the storeItem was created, and the second one is when the discount was applied

There will also be a new entry when the discount expires

==========================================================

###  
POST http://localhost:8080/store-item  
*Content-Type*: application/json  
  
{  
 "storeId": 0,  
 "itemId": 0,  
 "totalPrice": 0,  
 "units": 0,  
 "currency": "RON"  
}

This endpoint is used to register an item as a storeItem

I used the url: http://localhost:8080/store-item

And body:

{

  "storeId": 2,

  "itemId": 30,

  "totalPrice": 1337,

  "units": 10,

  "currency": "RON"

}

And the response is:

{

    "storeItemId": 77,

    "storeId": 2,

    "itemId": 30,

    "totalPrice": 1337.0,

    "units": 10.0,

    "currency": "RON"

}

==========================================================

GET http://localhost:8080/store-item/{{id}}  
*Content-Type*: application/json  
  
{  
 "storeId": 0,  
 "itemId": 0,  
 "totalPrice": 0,  
 "units": 0,  
 "currency": "RON"  
}

This endpoint is used to get a storeItem by their id

I’ll get the item that we entered in store 2 by using the url: http://localhost:8080/store-item/77

And the response is:

{

    "storeItemId": 77,

    "storeId": 2,

    "itemId": 30,

    "totalPrice": 1337.0,

    "units": 10.0,

    "currency": "RON"

}

==========================================================

PATCH http://localhost:8080/store-item/{{id}}  
*Content-Type*: application/json

This endpoint is used to update the storeItem

I used the url: http://localhost:8080/store-item/77

And body:

{

  "storeId": 2,

  "itemId": 30,

  "totalPrice": 7331,

  "units": 3,

  "currency": "RON"

}

And the response is:

{

    "storeItemId": 77,

    "storeId": 2,

    "itemId": 30,

    "totalPrice": 7331.0,

    "units": 3.0,

    "currency": "RON"

}

==========================================================

DELETE http://localhost:8080/store-item/{{id}}  
*Content-Type*: application/json  
  
{  
 "storeId": 0,  
 "itemId": 0,  
 "totalPrice": 0,  
 "units": 0,  
 "currency": "RON"  
}

This endpoint is used to delete a storeItem

I can’t delete it since we have an entry for that item in the price history

==========================================================

GET http://localhost:8080/store-item/store/{{storeId}}

This endpoint is used to get all the storeItems of a store

I used the url: http://localhost:8080/store-item/store/1

And the response is:

[

    {

        "storeItemId": 1,

        "storeId": 1,

        "itemId": 1,

        "totalPrice": 9.9,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 2,

        "storeId": 1,

        "itemId": 2,

        "totalPrice": 11.5,

        "units": 0.4,

        "currency": "RON"

    },

    {

        "storeItemId": 3,

        "storeId": 1,

        "itemId": 3,

        "totalPrice": 13.2,

        "units": 10.0,

        "currency": "RON"

    },

    {

        "storeItemId": 4,

        "storeId": 1,

        "itemId": 4,

        "totalPrice": 12.8,

        "units": 0.3,

        "currency": "RON"

    },

    {

        "storeItemId": 5,

        "storeId": 1,

        "itemId": 5,

        "totalPrice": 3.5,

        "units": 500.0,

        "currency": "RON"

    },

    {

        "storeItemId": 6,

        "storeId": 1,

        "itemId": 6,

        "totalPrice": 6.8,

        "units": 250.0,

        "currency": "RON"

    },

    {

        "storeItemId": 7,

        "storeId": 1,

        "itemId": 7,

        "totalPrice": 28.5,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 8,

        "storeId": 1,

        "itemId": 8,

        "totalPrice": 5.8,

        "units": 500.0,

        "currency": "RON"

    },

    {

        "storeItemId": 9,

        "storeId": 1,

        "itemId": 9,

        "totalPrice": 4.4,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 10,

        "storeId": 1,

        "itemId": 10,

        "totalPrice": 5.2,

        "units": 2.0,

        "currency": "RON"

    },

    {

        "storeItemId": 11,

        "storeId": 1,

        "itemId": 11,

        "totalPrice": 6.1,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 12,

        "storeId": 1,

        "itemId": 12,

        "totalPrice": 9.2,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 13,

        "storeId": 1,

        "itemId": 13,

        "totalPrice": 7.1,

        "units": 0.2,

        "currency": "RON"

    },

    {

        "storeItemId": 14,

        "storeId": 1,

        "itemId": 14,

        "totalPrice": 22.4,

        "units": 0.25,

        "currency": "RON"

    },

    {

        "storeItemId": 15,

        "storeId": 1,

        "itemId": 15,

        "totalPrice": 49.9,

        "units": 2.5,

        "currency": "RON"

    },

    {

        "storeItemId": 16,

        "storeId": 1,

        "itemId": 16,

        "totalPrice": 17.8,

        "units": 400.0,

        "currency": "RON"

    },

    {

        "storeItemId": 17,

        "storeId": 1,

        "itemId": 17,

        "totalPrice": 18.9,

        "units": 10.0,

        "currency": "RON"

    },

    {

        "storeItemId": 18,

        "storeId": 1,

        "itemId": 18,

        "totalPrice": 6.0,

        "units": 50.0,

        "currency": "RON"

    },

    {

        "storeItemId": 21,

        "storeId": 1,

        "itemId": 21,

        "totalPrice": 13.99,

        "units": 2.5,

        "currency": "RON"

    },

    {

        "storeItemId": 22,

        "storeId": 1,

        "itemId": 22,

        "totalPrice": 3.49,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 23,

        "storeId": 1,

        "itemId": 23,

        "totalPrice": 2.95,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 24,

        "storeId": 1,

        "itemId": 24,

        "totalPrice": 3.1,

        "units": 1.0,

        "currency": "RON"

    },

    {

        "storeItemId": 25,

        "storeId": 1,

        "itemId": 25,

        "totalPrice": 22.0,

        "units": 0.2,

        "currency": "RON"

    },

    {

        "storeItemId": 19,

        "storeId": 1,

        "itemId": 19,

        "totalPrice": 21.15,

        "units": 0.75,

        "currency": "RON"

    },

    {

        "storeItemId": 20,

        "storeId": 1,

        "itemId": 20,

        "totalPrice": 3.7050002,

        "units": 100.0,

        "currency": "RON"

    }

]

==========================================================

GET http://localhost:8080/store-item/{{storeItemId}}/best-value

This endpoint solves one of the requirements:

Product Substitutes & Recommendations: - Highlight "value per unit" (e.g., price per kg, price per liter) to help identify the best buys, even if the pack size differs.

Let’s test it on 3 storeItems:

http://localhost:8080/store-item/1/best-value

The response is:

{

    "storeItemId": 1,

    "storeId": 1,

    "itemId": 1,

    "totalPrice": 9.9,

    "units": 1.0,

    "valuePerUnit": 9.9,

    "currency": "RON"

}

http://localhost:8080/store-item/2/best-value

The response is:

{

    "storeItemId": 2,

    "storeId": 1,

    "itemId": 2,

    "totalPrice": 11.5,

    "units": 0.4,

    "valuePerUnit": 28.75,

    "currency": "RON"

}

http://localhost:8080/store-item/3/best-value

The response is:

{

    "storeItemId": 28,

    "storeId": 2,

    "itemId": 3,

    "totalPrice": 12.2,

    "units": 10.0,

    "valuePerUnit": 1.22,

    "currency": "RON"

}

==========================================================

GET http://localhost:8080/users/byId/{{userId}}  
*Content-Type*: application/json  
  
{  
 "firstName": "",  
 "lastName": "",  
 "email": ""  
}

This endpoint is used to get a user by their id

The url: http://localhost:8080/users/byId/1

The response:

{

    "userId": 1,

    "firstName": "MyTestingUser",

    "lastName": "AccesaInternship",

    "email": "testing\_the\_app@test.com",

    "spending": [

        {

            "spendingId": 1,

            "userId": 1,

            "storeId": 3,

            "purchasedItems": [

                {

                    "id": 1,

                    "spendingId": 1,

                    "itemName": "suc pepsi",

                    "pricePerUnit": 13.99,

                    "units": 1.0,

                    "unitType": "L"

                },

                {

                    "id": 2,

                    "spendingId": 1,

                    "itemName": "lapte zuzu",

                    "pricePerUnit": 10.1,

                    "units": 1.0,

                    "unitType": "L"

                },

                {

                    "id": 3,

                    "spendingId": 1,

                    "itemName": "piept pui",

                    "pricePerUnit": 27.9,

                    "units": 1.0,

                    "unitType": "KG"

                },

                {

                    "id": 4,

                    "spendingId": 1,

                    "itemName": "banane",

                    "pricePerUnit": 6.2,

                    "units": 1.0,

                    "unitType": "KG"

                }

            ],

            "totalPrice": 58.19,

            "purchaseDate": "2025-05-01T11:37:26"

        }

    ],

    "baskets": [

        {

            "basketId": 1,

            "userId": 1,

            "storeId": 1,

            "items": [

                {

                    "id": 1,

                    "basketId": 1,

                    "storeItemId": 1,

                    "quantity": 1.0,

                    "priceAtAddition": 9.9

                },

                {

                    "id": 2,

                    "basketId": 1,

                    "storeItemId": 2,

                    "quantity": 1.0,

                    "priceAtAddition": 11.5

                },

                {

                    "id": 3,

                    "basketId": 1,

                    "storeItemId": 3,

                    "quantity": 1.0,

                    "priceAtAddition": 13.2

                },

                {

                    "id": 4,

                    "basketId": 1,

                    "storeItemId": 4,

                    "quantity": 1.0,

                    "priceAtAddition": 12.8

                },

                {

                    "id": 5,

                    "basketId": 1,

                    "storeItemId": 5,

                    "quantity": 1.0,

                    "priceAtAddition": 3.5

                }

            ]

        },

        {

            "basketId": 2,

            "userId": 1,

            "storeId": 2,

            "items": [

                {

                    "id": 6,

                    "basketId": 2,

                    "storeItemId": 26,

                    "quantity": 1.0,

                    "priceAtAddition": 12.9

                },

                {

                    "id": 7,

                    "basketId": 2,

                    "storeItemId": 28,

                    "quantity": 1.0,

                    "priceAtAddition": 12.2

                },

                {

                    "id": 8,

                    "basketId": 2,

                    "storeItemId": 30,

                    "quantity": 1.0,

                    "priceAtAddition": 3.5

                },

                {

                    "id": 9,

                    "basketId": 2,

                    "storeItemId": 32,

                    "quantity": 1.0,

                    "priceAtAddition": 26.5

                },

                {

                    "id": 10,

                    "basketId": 2,

                    "storeItemId": 34,

                    "quantity": 1.0,

                    "priceAtAddition": 4.4

                },

                {

                    "id": 11,

                    "basketId": 2,

                    "storeItemId": 36,

                    "quantity": 1.0,

                    "priceAtAddition": 6.1

                },

                {

                    "id": 12,

                    "basketId": 2,

                    "storeItemId": 38,

                    "quantity": 1.0,

                    "priceAtAddition": 7.1

                }

            ]

        },

        {

            "basketId": 3,

            "userId": 1,

            "storeId": 3,

            "items": [

                {

                    "id": 13,

                    "basketId": 3,

                    "storeItemId": 50,

                    "quantity": 1.0,

                    "priceAtAddition": 13.1

                },

                {

                    "id": 14,

                    "basketId": 3,

                    "storeItemId": 51,

                    "quantity": 1.0,

                    "priceAtAddition": 3.6

                },

                {

                    "id": 15,

                    "basketId": 3,

                    "storeItemId": 52,

                    "quantity": 1.0,

                    "priceAtAddition": 7.0

                },

                {

                    "id": 16,

                    "basketId": 3,

                    "storeItemId": 53,

                    "quantity": 7.0,

                    "priceAtAddition": 27.9

                }

            ]

        }

    ],

    "priceAlerts": [

        {

            "priceAlertId": 1,

            "userId": 1,

            "storeItemId": 7,

            "targetPrice": 28.0

        },

        {

            "priceAlertId": 2,

            "userId": 1,

            "storeItemId": 21,

            "targetPrice": 12.5

        },

        {

            "priceAlertId": 3,

            "userId": 1,

            "storeItemId": 25,

            "targetPrice": 22.0

        }

    ]

}

==========================================================

###  
GET http://localhost:8080/users  
*Content-Type*: application/json  
  
{  
 "firstName": "",  
 "lastName": "",  
 "email": "",  
 "password": ""  
}

This url is used to get all the users that exist in our application

I won’t test the endpoint since we only have one user

==========================================================

POST http://localhost:8080/price-alerts  
*Content-Type*: application/json  
  
{  
 "userId": 0,  
 "storeItemId": 0,  
 "targetPrice": 0  
}

This endpoint is used to create a price alert for a storeItem

I used this url: http://localhost:8080/price-alerts

And this body:

{

  "userId": 1,

  "storeItemId": 6,

  "targetPrice": 5.5

}

And I received this response:

{

    "priceAlertId": 4,

    "userId": 1,

    "storeItemId": 6,

    "targetPrice": 5.5

}

==========================================================

PATCH http://localhost:8080/price-alerts/{{id}}  
*Content-Type*: application/json

This endpoint is used to update a price alert

The url that I used: http://localhost:8080/price-alerts/4

And the body:

{

  "userId": 1,

  "storeItemId": 6,

  "targetPrice": 6

}

And the response that I got:

{

    "priceAlertId": 4,

    "userId": 1,

    "storeItemId": 6,

    "targetPrice": 6.0

}

==========================================================

DELETE http://localhost:8080/price-alerts/{{id}}  
*Content-Type*: application/json  
  
{  
 "targetPrice": 0  
}

This endpoint is used to delete a price alert

The url that I used: http://localhost:8080/price-alerts/4

The response that I got:

{

    "priceAlertId": 4,

    "userId": 1,

    "storeItemId": 6,

    "targetPrice": 6.0

}

==========================================================

GET http://localhost:8080/price-alerts/user/{{userId}}

This endpoint is used to get all the alerts that a user has

The url that I used: http://localhost:8080/price-alerts/user/1

And the response that I received:

[

    {

        "priceAlertId": 1,

        "userId": 1,

        "storeItemId": 7,

        "targetPrice": 28.0

    },

    {

        "priceAlertId": 2,

        "userId": 1,

        "storeItemId": 21,

        "targetPrice": 12.5

    },

    {

        "priceAlertId": 3,

        "userId": 1,

        "storeItemId": 25,

        "targetPrice": 22.0

    }

]

==========================================================

GET http://localhost:8080/price-alerts/user/{{userId}}/get-triggered-alerts

This endpoint is used to solve one of the requirements:

Custom Price Alert: - Allow users to set a target price for a product. The system should be able to identify when a product's price drops to or below that target.

Let’s test this feature

The url that I use: http://localhost:8080/price-alerts/user/1/get-triggered-alerts

The response that I get:

[

    {

        "storeItem": {

            "storeItemId": 32,

            "storeId": 2,

            "itemId": 7,

            "totalPrice": 26.5,

            "units": 1.0,

            "currency": "RON"

        },

        "targetPrice": 28.0

    },

    {

        "storeItem": {

            "storeItemId": 25,

            "storeId": 1,

            "itemId": 25,

            "totalPrice": 22.0,

            "units": 0.2,

            "currency": "RON"

        },

        "targetPrice": 22.0

    }

]

We can see that the target for the first storeItem is 28, and the actual price of the storeItem is 26.5

For the second item, the target price is == totalPrice, so the targetPrice is active

Let’s change the price of the second storeItem:

I used the url: http://localhost:8080/store-item/25

And the body:

{

  "storeId": 1,

  "itemId": 25,

  "totalPrice": 50,

  "units": 0.2,

  "currency": "RON"

}

The response that I got:

{

    "storeItemId": 25,

    "storeId": 1,

    "itemId": 25,

    "totalPrice": 50.0,

    "units": 0.2,

    "currency": "RON"

}

Now if we run again the get triggered alerts url:

http://localhost:8080/price-alerts/user/1/get-triggered-alerts

This is the response that we receive:

[

    {

        "storeItem": {

            "storeItemId": 32,

            "storeId": 2,

            "itemId": 7,

            "totalPrice": 26.5,

            "units": 1.0,

            "currency": "RON"

        },

        "targetPrice": 28.0

    }

]

As you can see, the feature works

==========================================================

POST http://localhost:8080/discounts  
*Content-Type*: application/json  
  
{  
 "storeItemId": 0,  
 "storeId": 0,  
 "oldPrice": 0,  
 "discountPercentage": 0,  
 "startDate": "",  
 "endDate": ""  
}

This endpoint is used to create a discount for a storeItem

I’ll create a discount for the item:

{

        "itemId": 3,

        "itemName": "oua marimea M",

        "category": "OUA",

        "brand": "facute de cocos",

        "unitType": "BUC"

}

with the body:

{

  "storeItemId": 3,

  "storeId": 1,

  "oldPrice": 13.2,

  "discountPercentage": 25,

  "startDate": "2025-05-10T08:30:00",

  "endDate": "2025-05-17T08:30:00"

}

And received the response:

{

    "itemDiscountId": 4,

    "storeItemId": 3,

    "storeId": 1,

    "oldPrice": 13.2,

    "discountPercentage": 25.0,

    "startDate": "2025-05-10T08:30:00",

    "endDate": "2025-05-17T08:30:00",

    "itemId": 3

}

And now if we make a request on that storeItem: http://localhost:8080/store-item/3

The response is:

{

    "storeItemId": 3,

    "storeId": 1,

    "itemId": 3,

    "totalPrice": 9.9,

    "units": 10.0,

    "currency": "RON"

}

As we can see, the discount has been successfully applied

==========================================================

GET http://localhost:8080/discounts/active

This endpoint is used to get all the discounts that are active

The response is:

[

    {

        "itemDiscountId": 1,

        "storeItemId": 19,

        "storeId": 1,

        "oldPrice": 23.5,

        "discountPercentage": 10.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 19

    },

    {

        "itemDiscountId": 2,

        "storeItemId": 20,

        "storeId": 2,

        "oldPrice": 3.9,

        "discountPercentage": 5.0,

        "startDate": "2025-05-12T08:00:00",

        "endDate": "2025-05-19T08:30:00",

        "itemId": 20

    },

    {

        "itemDiscountId": 4,

        "storeItemId": 3,

        "storeId": 1,

        "oldPrice": 13.2,

        "discountPercentage": 25.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 3

    }

==========================================================

GET http://localhost:8080/discounts/all-items-discounts-with-max-discount-per-item

This endpoint is used to solve one of the requirements:

Best Discounts: - List products with the highest current percentage discounts across all tracked stores.

The backend gets all the store items that have a discount, and sort them descending, and return the list of discounts with unique items (the items with the lowest prices after the discounts)

If we call the url: http://localhost:8080/discounts/all-items-discounts-with-max-discount-per-item

The response is:

[

    {

        "itemDiscountId": 4,

        "storeItemId": 3,

        "storeId": 1,

        "oldPrice": 13.2,

        "discountPercentage": 25.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 3

    },

    {

        "itemDiscountId": 1,

        "storeItemId": 19,

        "storeId": 1,

        "oldPrice": 23.5,

        "discountPercentage": 10.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 19

    },

    {

        "itemDiscountId": 2,

        "storeItemId": 20,

        "storeId": 2,

        "oldPrice": 3.9,

        "discountPercentage": 5.0,

        "startDate": "2025-05-12T08:00:00",

        "endDate": "2025-05-19T08:30:00",

        "itemId": 20

    }

]

We get all the items that have a discount, but I want to add a bigger discount for storeItem with id 3, so we can see how it works:

I made a post: http://localhost:8080/discounts

With body:

{

  "storeItemId": 49,

  "storeId": 3,

  "oldPrice": 13.5,

  "discountPercentage": 30,

  "startDate": "2025-05-10T08:30:00",

  "endDate": "2025-05-17T08:30:00"

}

It is the same item, but in a different store.

The response is:

{

    "itemDiscountId": 5,

    "storeItemId": 49,

    "storeId": 3,

    "oldPrice": 13.5,

    "discountPercentage": 30.0,

    "startDate": "2025-05-10T08:30:00",

    "endDate": "2025-05-17T08:30:00",

    "itemId": 3

}

And now, if we call the url, we get:

[

    {

        "itemDiscountId": 6,

        "storeItemId": 49,

        "storeId": 3,

        "oldPrice": 13.5,

        "discountPercentage": 30.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 3

    },

    {

        "itemDiscountId": 1,

        "storeItemId": 19,

        "storeId": 1,

        "oldPrice": 23.5,

        "discountPercentage": 10.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 19

    },

    {

        "itemDiscountId": 2,

        "storeItemId": 20,

        "storeId": 2,

        "oldPrice": 3.9,

        "discountPercentage": 5.0,

        "startDate": "2025-05-12T08:00:00",

        "endDate": "2025-05-19T08:30:00",

        "itemId": 20

    }

]

So for discounts for the same item in the multiple stores, the endpoint returns the one that has the biggest discount (for all items), and we have the list of discounts

==========================================================

GET http://localhost:8080/discounts/recent?  
 hours={{$random.integer(100)}}

This endpoint is used to solve one of the requirements:

New Discounts: - List discounts that have been newly added (e.g., within the last 24 hours, 48 and 72)

The backend checks for the discounts that have became active within a specific period of time and returns them (startDate <= now <= startDate+numberOfHours)

Let’s test it with some urls:  
Discounts added within the last 24 hours: http://localhost:8080/discounts/recent?hours=24

The response is:

[]

There are no discounts added in the last 24 hours

For the ones in the last 48 hours: http://localhost:8080/discounts/recent?hours=48

The response is:

[

    {

        "itemDiscountId": 2,

        "storeItemId": 20,

        "storeId": 2,

        "oldPrice": 3.9,

        "discountPercentage": 5.0,

        "startDate": "2025-05-12T08:00:00",

        "endDate": "2025-05-19T08:30:00",

        "itemId": 20

    }

]

For the 72, the response still remains the same

And for the 128 hours we get:

[

    {

        "itemDiscountId": 1,

        "storeItemId": 19,

        "storeId": 1,

        "oldPrice": 23.5,

        "discountPercentage": 10.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 19

    },

    {

        "itemDiscountId": 2,

        "storeItemId": 20,

        "storeId": 2,

        "oldPrice": 3.9,

        "discountPercentage": 5.0,

        "startDate": "2025-05-12T08:00:00",

        "endDate": "2025-05-19T08:30:00",

        "itemId": 20

    },

    {

        "itemDiscountId": 4,

        "storeItemId": 3,

        "storeId": 1,

        "oldPrice": 13.2,

        "discountPercentage": 25.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 3

    },

    {

        "itemDiscountId": 5,

        "storeItemId": 28,

        "storeId": 2,

        "oldPrice": 12.2,

        "discountPercentage": 25.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

        "itemId": 3

    },

    {

        "itemDiscountId": 6,

        "storeItemId": 49,

        "storeId": 3,

        "oldPrice": 13.5,

        "discountPercentage": 30.0,

        "startDate": "2025-05-10T08:30:00",

        "endDate": "2025-05-17T08:30:00",

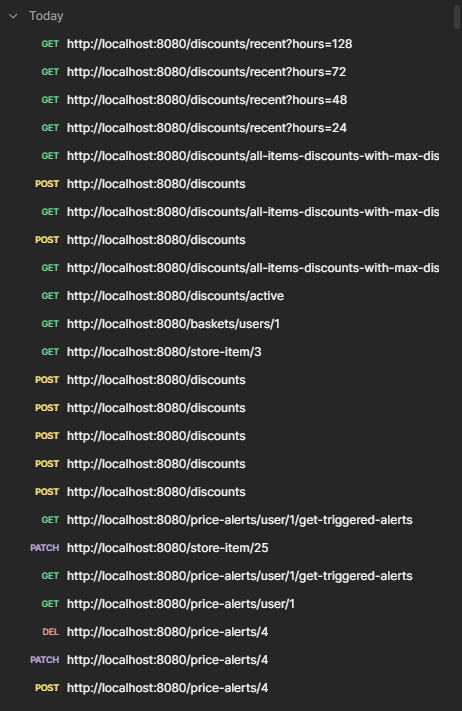
        "itemId": 3

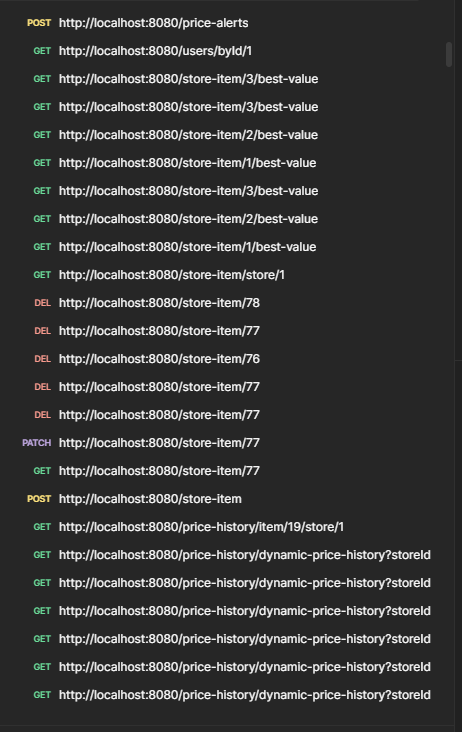
    }

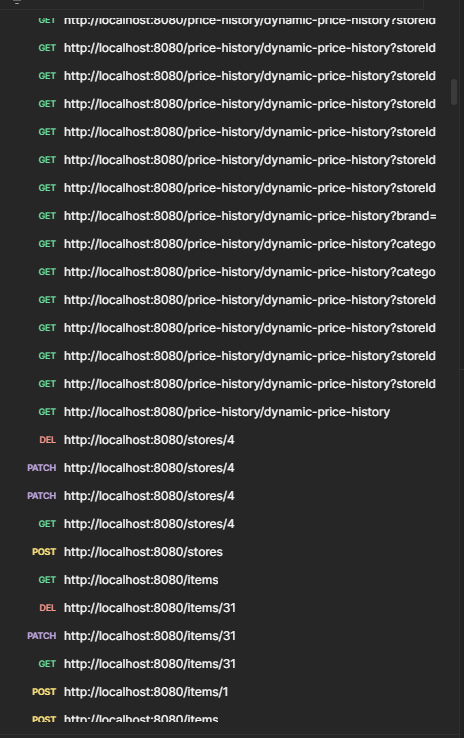
]

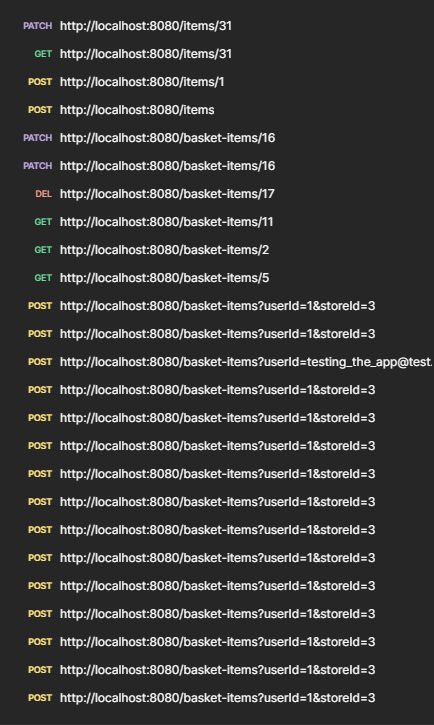
These dates are relative to the date that I’m calling the endpoints right now: 14.05.2025 2:28 AM

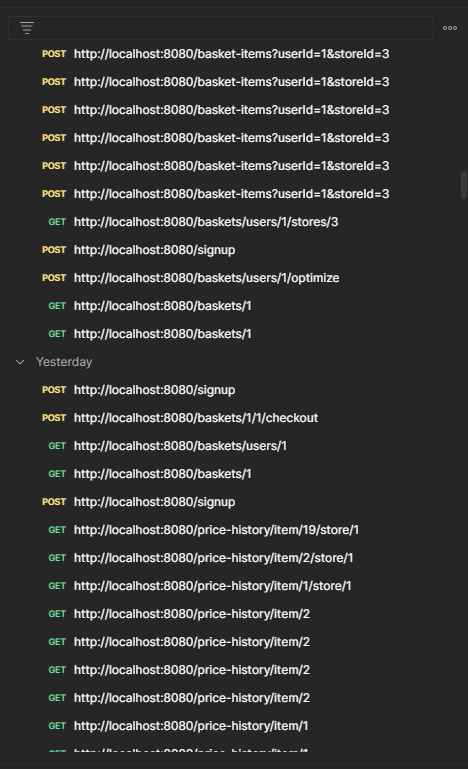
Proof that I really did all the testing:











Moldovan Ilie-Gabriel

End of the API documentation for my Accesa Internship Coding Challenge application