

Sprawozdanie

Programowanie Bazodanowe

Laboratorium 6

Tomasz Richert

BII_EF

c. Pobranie listy produktów z możliwością:

- Obustronnego sortowania, po nazwie, cenie lub nazwie grupy.
- Funkcja powinna zwracać id, nazwę, cenę, nazwę grupy.
- Filtrowania po nazwie.
- Filtrowania po nazwie grupy.
- Filtrowania po id grupy.
- Funkcja powinna zwracać tylko aktywne produkty z możliwością wymuszenia pobrania nieaktywnych.

The screenshot shows a REST client interface with the following sections:

- GET /api/Products**: The endpoint being tested.
- Parameters**: A table of query parameters.

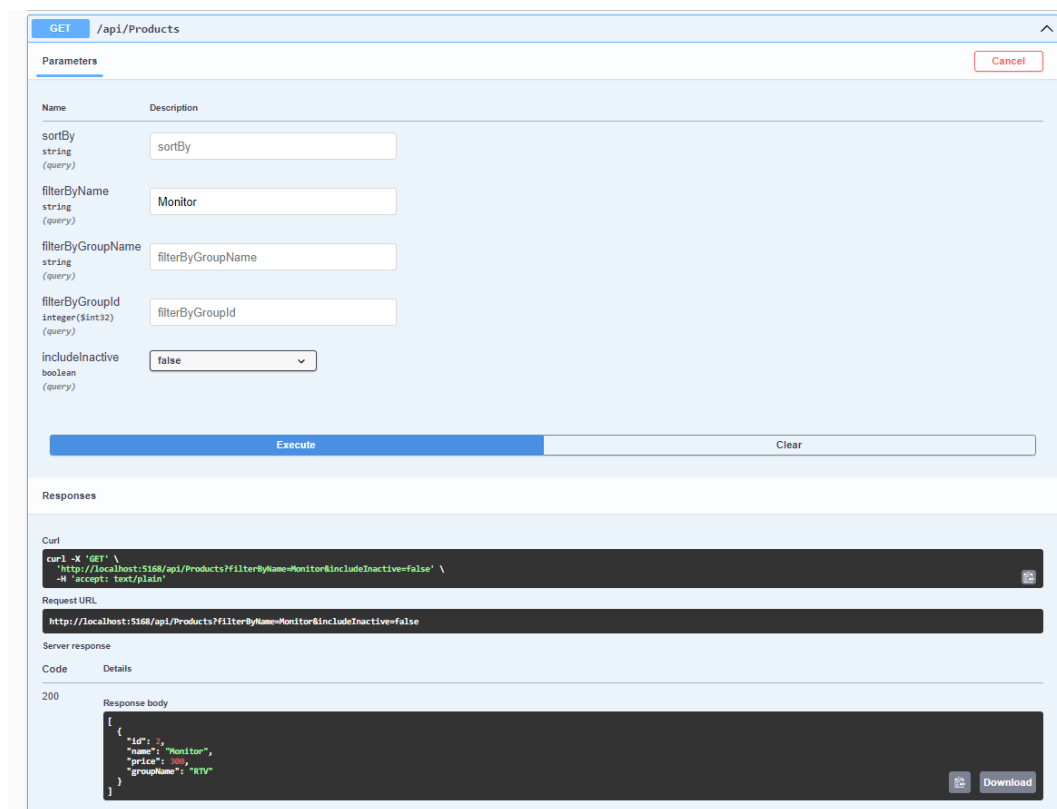
Name	Description
sortBy string (query)	sortBy
filterByName string (query)	filterByName
filterByGroupName string (query)	filterByGroupName
filterByGroupid integer(\$int32) (query)	filterByGroupid
includeInactive boolean (query)	false
- Execute**: A blue button to run the request.
- Clear**: A button to clear the parameters.
- Responses**:
 - Curl**:

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?includeInactive=false' \
  -H 'accept: text/plain'
```
 - Request URL**:

```
http://localhost:5168/api/Products?includeInactive=false
```
 - Server response**:

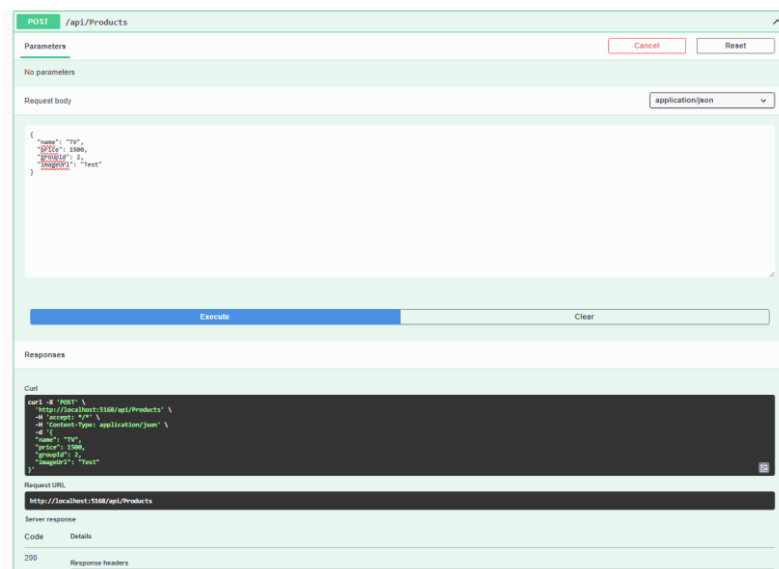
Code	Details
200	<div>Response body</div> <pre>{ { "id": 1, "name": "Fridge", "price": 1200, "groupname": "AGD" }, { "id": 2, "name": "Monitor", "price": 300, "groupname": "RTV" } }</pre> <div>Download</div>

Rysunek 1 Bez parametrów



Rysunek 2 Z parametrem FilterByName

- d. Dodanie nowego produktu, gdzie użytkownik podaje nazwę, cenę oraz id grupy.



Rysunek 3 Operacja post

Products

GET

/api/Products

Parameters

Cancel

Name	Description
sortBy string (query)	<input type="text" value="sortBy"/>
filterByName string (query)	<input type="text" value="TV"/>
filterByGroupName string (query)	<input type="text" value="filterByGroupName"/>
filterById integer(\$int32) (query)	<input type="text" value="filterById"/>
includeInactive boolean (query)	<input type="text" value="false"/>

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?filterByName=TV&includeInactive=false' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5168/api/Products?filterByName=TV&includeInactive=false
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 1, "name": "TV", "price": 1500, "groupName": "RTV" }</pre></div><div>Download</div></div>

Response headers

Rysunek 4 Weryfikacja dodanego produktu, dzięki pierwszemu endpointowi.

e. Dezaktywację lub usunięcie produktu.

PATCH

/api/Products

Parameters

Cancel

Name	Description
productId integer(\$int32) (query)	<input type="text" value="1"/>

ExecuteClear

Responses

Curl

```
curl -X 'PATCH' \
  'http://localhost:5168/api/Products?productId=1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Products?productId=1
```

Server response

Code	Details						
200	<div><div>Response headers</div><div><pre>content-length: 0 date: Wed, 27 Mar 2024 16:51:05 GMT server: Kestrel</pre></div><div>Responses</div><table><thead><tr><th>Code</th><th>Description</th><th>Links</th></tr></thead><tbody><tr><td>200</td><td>Success</td><td>No links</td></tr></tbody></table></div>	Code	Description	Links	200	Success	No links
Code	Description	Links					
200	Success	No links					

Rysunek 5 Dezaktywacja

Przed

(query)

includeInactive

boolean

(query)

Execute

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?includeInactive=true' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5168/api/Products?includeInactive=true
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 1, "name": "Fridge", "price": 1200, "groupName": "AGD" }, { "id": 2, "name": "Monitor", "price": 300, "groupName": "RTV" }, { "id": 4, "name": "TV", "price": 1500, "groupName": "RTV" }]</pre>

Po

(query)

includeInactive

boolean

(query)

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?includeInactive=false' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5168/api/Products?includeInactive=false
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 2, "name": "Monitor", "price": 300, "groupname": "RTV" }, { "id": 4, "name": "TV", "price": 1500, "groupname": "RTV" }]</pre> <p>Download</p>

DELETE

/api/Products

Parameters

Cancel

Name	Description
id	
integer(\$int32)	1
(query)	

Execute

Responses

Code	Description	Links
200	Success	No links

Rysunek 6 Usunięcie Produktu

Przed

Response body

```
[
  {
    "id": 1,
    "name": "Fridge",
    "price": 1500,
    "groupName": "AGD"
  },
  {
    "id": 2,
    "name": "TV",
    "price": 1300,
    "groupName": "RTV"
  },
  {
    "id": 3,
    "name": "Dishwasher",
    "price": 900,
    "groupName": "AGD"
  }
]
```

Po

Code	Details
200	<div>Response body</div> <pre>[{ "id": 2, "name": "TV", "price": 1300, "groupName": "RTV" }, { "id": 3, "name": "Dishwasher", "price": 900, "groupName": "AGD" }]</pre>

f. Aktywację produktu.

PATCH

/api/Products/ActivateProduct

Cancel

Parameters

Name	Description
productId	
integer(\$int32)	2
(query)	

ExecuteClear

Responses

Curl

curl -X 'PATCH' \

'http://localhost:5168/api/Products/ActivateProduct?productId=2' \

-H 'accept: */*' \

-d ''

Request URL

http://localhost:5168/api/Products/ActivateProduct?productId=2

Server response

Code	Details
200	<div>Response headers</div> <div>content-length: 0</div> <div>date: Wed, 27 Mar 2024 17:11:26 GMT</div> <div>server: Kestrel</div>

Responses

Code	Description	Links
200	Success	No links

g. Dodanie produktu do koszyka.

Basket

POST

/api/Basket/AddProductToBasket

Cancel

Parameters

Name	Description
userId	
integer(\$int32)	1
(query)	
productId	
integer(\$int32)	2
(query)	

ExecuteClear

Responses

Curl

curl -X 'POST' \

'http://localhost:5168/api/Basket/AddProductToBasket?userId=1&productId=2' \

-H 'accept: */*' \

-d ''

Request URL

http://localhost:5168/api/Basket/AddProductToBasket?userId=1&productId=2

Server response

Code	Details
200	<div>Response headers</div> <div>content-length: 0</div> <div>date: Wed, 27 Mar 2024 17:45:49 GMT</div> <div>server: Kestrel</div>

Responses

Code	Description	Links
200		No links

Rysunek 7 Dodanie produktu do koszyka

h. Zmianę ilości produktu w koszyku.

PATCH/api/Basket/ChangeBasketPositionQuantity

Parameters

Name

Description

basketPositionId

integer(\$int32)

(query)

1

newQuantity

integer(\$int32)

(query)

3

Execute

Clear

Responses

Curl

```
curl -X 'PATCH' \
  'http://localhost:5168/api/Basket/ChangeBasketPositionQuantity?basketPositionId=1&newQuantity=3' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Basket/ChangeBasketPositionQuantity?basketPositionId=1&newQuantity=3
```

Server response

Code

Details

200

Response headers

content-length: 0

date: Wed, 27 Mar 2024 17:46:29 GMT

server: Kestrel

Responses

Code

Description

Links

200

Success

No links

Rysunek 8 Zmiana ilości produktu w koszyku

dbo.BasketPositions [Data]				
Max Rows: 1000				
	ID	ProductID	UserID	Amount
	1	2	1	3
	NULL	NULL	NULL	NULL

Rysunek 9 Potwierdzenie zmiany w bazie danych

i. Usunięcie produktu z koszyka.

DELETE /api/Basket/RemoveProductFromBasket

Parameters

Cancel

Name	Description
id	
integer(\$int32)	2
(query)	

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5168/api/Basket/RemoveProductFromBasket?id=2' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Basket/RemoveProductFromBasket?id=2
```

Server response

Code	Details
200	<div><div>Response headers</div><div><pre>content-length: 0 date: Wed, 27 Mar 2024 18:25:59 GMT server: Kestrel</pre></div></div>

Responses

Rysunek 10 Usunięcie produktów z koszyka

j. Wygenerowanie zamówienia wraz z pozycjami z koszyka.

Orders

POST

/api/Orders/GenerateOrder

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	

ExecuteClear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:5168/api/Orders/GenerateOrder?userId=1' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://localhost:5168/api/Orders/GenerateOrder?userId=1
```

Server response

Code	Details
200	<div>Response headers<pre>content-length: 0 date: Wed, 27 Mar 2024 18:19:36 GMT server: Kestrel</pre></div>

Responses

Code	Description	Links
------	-------------	-------

Rysunek 11 Wygenerowanie zamówienia

	ID	OrderID	Amount	Price	ProductID
	9	9	3	3900	2
	NULL	NULL	NULL	NULL	NULL

Rysunek 12 Potwierdzenie wygenerowania zamówienia

k. Opłacenie zamówienia (podajemy wpłacaną kwotę).

PATCH /api/Orders/PayOrder

Parameters Cancel

Name	Description
orderId	
integer(\$int32)	9
(query)	
amountPaid	
number(\$double)	3900
(query)	

Execute Clear

Responses

Curl

```
curl -X 'PATCH' \
'http://localhost:5168/api/Orders/PayOrder?orderId=9&amountPaid=3900' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Orders/PayOrder?orderId=9&amountPaid=3900
```

Server response

Code	Details
200	<div>Response headers</div> <div>content-length: 0 date: Wed, 27 Mar 2024 18:23:25 GMT server: Kestrel</div>

Rysunek 13 Opłacenie zamówienia

dbo.Orders [Data] | dbo.OrderPositions [Data] | dbo.Products [Design]

Max Rows: 1000

	ID	UserID	Date	IsPaid
▶	3	1	27.03.2024 19:0...	False
	4	1	27.03.2024 19:0...	False
	5	1	27.03.2024 19:1...	False
	6	1	27.03.2024 19:1...	False
	7	1	27.03.2024 19:1...	False
	8	1	27.03.2024 19:1...	False
	9	1	27.03.2024 19:1...	True
⚙	NULL	NULL	NULL	NULL

Rysunek 14 Potwierdzenie zapłaty

BLL_DB

c. Pobranie listy produktów z możliwością:

- Obustronnego sortowania, po nazwie, cenie lub nazwie grupy.
- Funkcja powinna zwracać id, nazwę, cenę, nazwę grupy.
- Filtrowania po nazwie.
- Filtrowania po nazwie grupy.
- Filtrowania po id grupy.
- Funkcja powinna zwracać tylko aktywne produkty z możliwością wymuszenia pobrania nieaktywnych.

GET /api/Products

Parameters

Cancel

Name	Description
sortBy string (query)	<input type="text" value="sortBy"/>
filterByName string (query)	<input type="text" value="filterByName"/>
filterByGroupName string (query)	<input type="text" value="filterByGroupName"/>
filterByGroupId integer(\$int32) (query)	<input type="text" value="filterByGroupId"/>
includeInactive boolean (query)	<input type="text" value="false"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?includeInactive=false' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5168/api/Products?includeInactive=false
```

Server response

Code	Details
200	<div><div>Response body</div><pre>{ "id": 1, "name": "Iodowka", "price": 1000, "groupName": "a" }</pre><div><div>Download</div></div></div>

GET /api/Products

Parameters

Name

Description

sortBy

string
(query)

sortBy

filterByName

string
(query)

filterByName

filterByGroupName

string
(query)

filterByGroupName

filterByGroupId

integer(int32)
(query)

filterByGroupId

includeInactive

boolean
(query)

true

Execute

Clear

Responses

Curl

curl -X 'GET' \n'http://localhost:5168/api/Products?includeInactive=true' \n-H 'accept: text/plain'

Request URL

http://localhost:5168/api/Products?includeInactive=true

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "name": "lodowka",
  "price": 1000,
  "groupName": "a"
},
{
  "id": 2,
  "name": "zamozarka",
  "price": 2222,
  "groupName": "b"
},
{
  "id": 3,
  "name": "telewizor",
  "price": 1111,
  "groupName": "a"
}

```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 14 May 2024 20:05:38 GMT
server: Restful
transfer-encoding: chunked

```

Responses

Code

Description

Links

d. Dodanie nowego produktu, gdzie użytkownik podaje nazwę, cenę oraz id grupy.

POST /api/Products/AddProduct

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "name": "testAdd",
  "price": 99,
  "groupId": 2
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5168/api/Products/AddProduct' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "testAdd",
    "price": 99,
    "groupId": 2
  }'
```

Wynik dodania

Curl

```
curl -X 'GET' \
  'http://localhost:5168/api/Products?includeInactive=false' \
  -H 'accept: text/plain'
```

Request URL

http://localhost:5168/api/Products?includeInactive=false

Server response

Code

Details

200

Response body

```
{
  {
    "id": 1,
    "name": "Iodowka",
    "price": 1800,
    "groupName": "a"
  },
  {
    "id": 2,
    "name": "samozazarka",
    "price": 2222,
    "groupName": "b"
  },
  {
    "id": 7,
    "name": "testAdd",
    "price": 99,
    "groupName": "b"
  },
  {
    "id": 8,
    "name": "testAdd",
    "price": 99,
    "groupName": "b"
  }
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 14 May 2024 20:09:01 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code

Description

Links

e. Dezaktywację lub usunięcie produktu.

PATCH

/api/Products/DeactivateProduct

Parameters

Name	Description
productid	<input type="text" value="2"/>

Execute

Responses

Code	Description	Links
200	Success	No links

2	zamrazarka	2222	asdad	False	2
---	------------	------	-------	-------	---

DELETE

/api/Products/DeleteProduct

Parameters

Name	Description
id	<input type="text" value="2"/>

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5168/api/Products/DeleteProduct?id=2' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Products/DeleteProduct?id=2
```

Server response

Code	Details
200	<div>Response headers<div>content-length: 0 date: Tue, 14 May 2024 20:11:35 GMT server: Kestrel</div></div>

Responses

Code	Description	Links
200	Success	No links

f. Aktywację produktu.

	ID	Name	Price	Image	IsActive	GroupID
▶	1	lódowka	1000	xyz	False	1

PATCH /api/Products/ActivateProduct

Parameters

Cancel

Name	Description
productid	<input type="text" value="1"/>
integer(\$int32)	(query)

ExecuteClear

Responses

Curl

```
curl -X 'PATCH' \
'http://localhost:5168/api/Products/ActivateProduct?productid=1' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5168/api/Products/ActivateProduct?productid=1
```

Server response

Code	Details
200	<div>Response headers<pre>content-length: 0 date: Tue, 14 May 2024 20:13:39 GMT server: Kestrel</pre></div>

Responses

Code	Description	Links
200	Success	No links

	ID	Name	Price	Image	IsActive	GroupID
▶	1	lódowka	1000	xyz	True	1

g. Dodanie produktu do koszyka.

POST

/api/Basket/AddProductToBasket

Parameters

Cancel

Name	Description
userId	
integer(\$int32)	1
(query)	
productId	
integer(\$int32)	1
(query)	

Execute

Clear

Responses

Curl

curl -X 'POST' \\\n'http://localhost:5168/api/Basket/AddProductToBasket?userId=1&productId=1' \\\n-H 'accept: */*' \\\n-d ''

Request URL

http://localhost:5168/api/Basket/AddProductToBasket?userId=1&productId=1

Server response

Code	Details
200	<div>Response headers</div> <div>content-length: 0\\n\\ndate: Tue, 14 May 2024 20:23:12 GMT\\n\\nservice: Kestrel</div>

Responses

Code	Description	Links
200	Success	No links

	ID	ProductID	UserID	Amount
	1	1	1	1

h. Zmianę ilości produktu w koszyku.

PATCH

/api/Basket/ChangeBasketPositionQuantity

Parameters

Cancel

Name	Description
basketPositionId	
integer(\$int32)	1
(query)	
newQuantity	
integer(\$int32)	2
(query)	

Execute

Clear

Responses

Curl

curl -X 'PATCH' \\\n'http://localhost:5168/api/Basket/ChangeBasketPositionQuantity?basketPositionId=1&newQuantity=2' \\\n-H 'accept: */*'

Request URL

http://localhost:5168/api/Basket/ChangeBasketPositionQuantity?basketPositionId=1&newQuantity=2

Server response

Code	Details
200	<div>Response headers</div> <div>content-length: 0\\n\\ndate: Tue, 14 May 2024 20:24:22 GMT\\n\\nservice: Kestrel</div>

Responses

Code	Description	Links
200	Success	No links

	ID	ProductID	UserID	Amount
	1	1	1	2
	NULL	NULL	NULL	NULL

iii. Linknię nowego produktu w koszyku

i. Usunięcie produktu z koszyka.

DELETE

/api/Basket/RemoveProductFromBasket

Parameters

Cancel

Name	Description
id	
integer(\$int32)	1
(query)	

Execute

Clear

Responses

Curl

curl -X 'DELETE' \
'http://localhost:5168/api/Basket/RemoveProductFromBasket?id=1' \
-H 'accept: */*'

Request URL

http://localhost:5168/api/Basket/RemoveProductFromBasket?id=1

Server response

Code	Details
200	<div>Response headers</div> <div> content-length: 0 date: Tue, 14 May 2024 20:24:56 GMT server: Kestrel </div>

Responses

Code	Description	Links
200	Success	No links

j. Wygenerowanie zamówienia wraz z pozycjami z koszyka.

POST /api/Orders/GenerateOrder

Parameters

Cancel

Name	Description
userId	integer(\$int32) (query)

1

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5168/api/Orders/GenerateOrder?userId=1' \
  -H 'accept: */*' \
  -d ''
```

Request URL

```
http://localhost:5168/api/Orders/GenerateOrder?userId=1
```

Server response

Code	Details
200	<div><div>Response headers</div><div><pre>content-length: 0 date: Tue, 14 May 2024 20:47:21 GMT server: Kestrel</pre></div></div>

Responses

Code	Description	Links
200	Success	No links

	ID	OrderID	Amount	Price	ProductID
	2	8	2	2000	1

k. Opłacenienie zamówienia (podajemy wpłacaną kwotę).

PATCH /api/Orders/PayOrder

Parameters

Name

Description

orderId

integer(\$int32)

(query)

8

amountPaid

number(\$double)

(query)

4000

Execute

Clear

Responses

Curl

curl -X 'PATCH' \
'http://localhost:5168/api/Orders/PayOrder?orderId=8&amountPaid=4000' \
-H 'accept: */*'

Request URL

http://localhost:5168/api/Orders/PayOrder?orderId=8&amountPaid=4000

Server response

Code

Details

200

Response headers

content-length: 0
date: Tue, 14 May 2024 20:58:32 GMT
server: Kestrel

	8	1	14.05.2024 22:47:21	True
	NULL	NULL	NULL	NULL

Wnioski

Obie metody mają swoje zalety i wady, a wybór między nimi powinien zależeć od specyficznych wymagań projektu. Entity Framework oferuje wyższy poziom abstrakcji, elastyczność oraz łatwość utrzymania kodu, co może być korzystne dla szybkiego rozwoju aplikacji. Z kolei procedury składowane oferują lepszą wydajność i większą kontrolę nad optymalizacją zapytań, co może być kluczowe w przypadku aplikacji wymagających intensywnego przetwarzania danych.

W praktyce często stosuje się hybrydowe podejście, gdzie Entity Framework jest używany do większości operacji CRUD, natomiast procedury składowane są wykorzystywane do skomplikowanych operacji, które wymagają optymalizacji wydajnościowej.